



日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office

出 願 年 月 日

Date of Application:

2000年 1月27日

出 願 番 号

Application Number:

特願2000-019172

ST.10/C]:

[JP2000-019172]

出 願 人

Applicant(s):

株式会社スクウェア

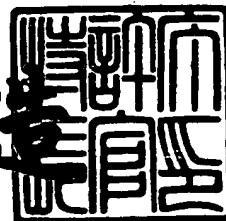
RECEIVED
SEP 13 2002
Technology Center 2600

CERTIFIED COPY OF
PRIORITY DOCUMENT

2002年 7月19日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造





COPY OF PAPERS
ORIGINALLY FILED

9672
PATENT #5
V.24
091802

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Sasaki, Ryusuke

Attorney Docket No.: SIP1P042

Application No.: 09/771,111

Examiner: Not yet assigned

Filed: January 26, 2001

Group: 2672

Title: METHODS AND APPARATUS FOR
TRANSFORMING THREE-DIMENSIONAL
OBJECTS IN VIDEO GAMES

RECEIVED
SEP 13 2002
Technology Center 2600

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as First Class Mail to: Commissioner for Patents, Washington, DC 20231 on September 5, 2002.

Signed:

Mia Mitchell-Haynes

TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENTS

Commissioner for Patents
Washington, D.C. 20231


Sir:

Transmitted herewith is the certified copy of the priority document for the above-referenced patent application.

The Commissioner is authorized to charge any fees that may be due to Deposit Account No. 500388 (Order No. SIP1P042).

Respectfully submitted,

BEYER WEAVER & THOMAS, LLP


Russell N. Swerdon

Registration No. 36,943

P.O. Box 778
Berkeley, CA 94704-0778

【書類名】 特許願

【整理番号】 JP0160

【あて先】 特許庁長官殿

【国際特許分類】 G06T 15/70

【発明者】

【住所又は居所】 東京都目黒区下目黒1丁目8番1号 株式会社スクウェア
アヴィジュアルワークス内

【氏名】 佐々木 隆典

【特許出願人】

【識別番号】 391049002

【氏名又は名称】 株式会社スクウェア

【代理人】

【識別番号】 100103528

【弁理士】

【氏名又は名称】 原田 一男

【手数料の表示】

【予納台帳番号】 076762

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9910116

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ビデオゲームにおける三次元オブジェクト変形方法及びビデオゲーム装置、並びにビデオゲーム用のプログラムを記録したコンピュータ読み取り可能な記録媒体

【特許請求の範囲】

【請求項 1】

複数の頂点により形状が定められる三次元オブジェクトを変形する、ビデオゲーム用のプログラムを記録したコンピュータ読み取り可能な記録媒体であって、前記コンピュータに、

複数のジョイントにより構成され、且つ前記複数の頂点の各々が前記複数のジョイントのいずれか 1 つに対応付けられている前記三次元オブジェクトの仮想スケルトンにおける各前記ジョイントの回転角を、当該仮想スケルトンの動きが定義されたアニメーション・データに基づいてフレーム表示周期毎に取得する第 1 ステップと、

前記第 1 ステップで取得された各前記ジョイントの回転角と当該ジョイントに対応する前記頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、当該回転角に従って前記頂点を前記フレーム表示周期毎に移動する第 2 ステップと、

を実行させるためのプログラムであることを特徴とするビデオゲーム用のコンピュータ読み取り可能な記録媒体。

【請求項 2】

前記第 2 ステップが、

前記第 1 ステップで取得された前記ジョイントの回転角により定まる 1 つの回転軸に対する 1 つの回転角と当該ジョイントに対応する前記頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、前記 1 つの回転軸に沿って、当該回転角だけ当該頂点を回転させて得られる位置に前記頂点を前記フレーム表示周期毎に移動するステップ

であることを特徴とする請求項 1 記載のビデオゲーム用のコンピュータ読み取り可能な記録媒体。

【請求項 3】

前記第 1 ステップにおいて、互いに直交する 3 つの回転軸毎の各前記ジョイントの回転角を取得し、

前記第 2 ステップが、

前記第 1 ステップで取得された前記ジョイントの前記 3 つの回転軸毎の回転角に基づき、一つの回転軸と一つの回転角とを計算する第 3 ステップと、

前記第 3 ステップで計算された前記一つの回転角を、前記頂点に予め定義された重みに従って補間し、当該頂点の回転角を計算する第 4 ステップと、

前記第 4 ステップで計算された回転角と前記一つの回転軸とに従って前記頂点を前記フレーム表示周期毎に移動させる第 5 ステップと、

を含むことを特徴とする請求項 1 記載のビデオゲーム用のコンピュータ読み取り可能な記録媒体。

【請求項 4】

前記第 4 ステップが、

前記一つの回転角を、前記頂点に予め定義された重みに従って球状線形補間し、当該頂点の回転角を計算するステップ

であることを特徴とする請求項 3 記載のビデオゲーム用のコンピュータ読み取り可能な記録媒体。

【請求項 5】

1 つの前記ジョイントに対応する頂点は当該頂点に予め定義された重みによりソートされており、

前記第 2 ステップは、

直前に移動された頂点に予め定義された重みと同じ重みが移動すべき頂点に定義されているか判断する第 3 ステップと、

前記第 3 ステップにおいて直前に移動された頂点に予め定義された重みと同じ重みが移動すべき頂点に定義されていないと判断された場合には、前記移動すべき頂点に対応する前記ジョイントの前記第 1 ステップで取得された回転角と当該移動すべき頂点に予め定義された重みに基づいて当該移動すべき頂点の回転角を算出し、当該回転角に関する情報を格納する第 4 ステップと、

前記第 4 ステップにおいて格納された回転角に従って、前記移動すべき頂点を移動させる第 5 ステップと、

前記第 3 ステップにおいて直前に移動された頂点に予め定義された重みと同じ重みが移動すべき頂点に定義されていると判断された場合には、前記第 5 ステップで格納された回転角に従って前記移動すべき頂点を移動させる第 6 ステップと

を含むことを特徴とする請求項 1 記載のビデオゲーム用のコンピュータ読み取り可能な記録媒体。

【請求項 6】

前記第 2 ステップは、

同一の前記ジョイントに対応した頂点のうち既に移動された頂点に予め定義されている重みと同一の重みが移動すべき頂点に定義されているか判断する第 3 ステップと、

前記第 3 ステップにおいて同一の前記ジョイントに対応した頂点のうち既に移動された頂点に予め定義されている重みと同一の重みが移動すべき頂点に定義されていないと判断された場合には、前記移動すべき頂点に対応する前記ジョイントの前記第 1 ステップで取得された回転角と当該移動すべき頂点に予め定義された重みとに従って当該頂点の回転角を算出し、当該回転角に関する情報を当該頂点の重みと関係付けて格納する第 4 ステップと、

前記第 4 ステップにおいて算出された回転角に従って、前記移動すべき頂点を移動させる第 5 ステップと、

前記第 3 ステップにおいて同一の前記ジョイントに対応した頂点のうち既に移動された頂点に予め定義されている重みと同一の重みが移動すべき頂点に定義されていると判断された場合には、前記移動すべき頂点に予め定義された重みに関係付けて前記第 4 ステップで格納された回転角を取得し、当該回転角に従って、当該移動すべき頂点を移動させる第 6 ステップと、

を含むことを特徴とする請求項 1 記載のビデオゲーム用のコンピュータ読み取り可能な記録媒体。

【請求項 7】

複数の頂点により形状が定められる三次元オブジェクトを変形する、ビデオゲームにおける三次元オブジェクト変形方法であって、

複数のジョイントにより構成され、且つ前記複数の頂点の各々が前記複数のジョイントのいずれか 1 つに対応付けられている前記三次元オブジェクトの仮想スケルトンにおける各前記ジョイントの回転角を、当該仮想スケルトンの動きが定義されたアニメーション・データに基づいてフレーム表示周期毎に取得する第 1 ステップと、

前記第 1 ステップで取得された各前記ジョイントの回転角と当該ジョイントに対応する前記頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、当該回転角に従って前記頂点を前記フレーム表示周期毎に移動する第 2 ステップと、

を含むことを特徴とするビデオゲームにおける三次元オブジェクト変形方法。

【請求項 8】

前記第 2 ステップが、

前記第 1 ステップで取得された前記ジョイントの回転角により定まる 1 つの回転軸に対する 1 つの回転角と当該ジョイントに対応する前記頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、前記 1 つの回転軸に沿って、当該回転角だけ当該頂点を回転させて得られる位置に前記頂点を前記フレーム表示周期毎に移動するステップ

であることを特徴とする請求項 7 記載のビデオゲームにおける三次元オブジェクト変形方法。

【請求項 9】

前記第 1 ステップにおいて、互いに直交する 3 つの回転軸毎の各前記ジョイントの回転角を取得し、

前記第 2 ステップが、

前記第 1 ステップで取得された前記ジョイントの前記 3 つの回転軸毎の回転角に基づき、一つの回転軸と一つの回転角とを計算する第 3 ステップと、

前記第 3 ステップで計算された前記一つの回転角を、前記頂点に予め定義された重みに従って補間し、当該頂点の回転角を計算する第 4 ステップと、

前記第 4 ステップで計算された回転角と前記一つの回転軸とに従って前記頂点を前記フレーム表示周期毎に移動させる第 5 ステップと、

を含むことを特徴とする請求項 1 記載のビデオゲームにおける三次元オブジェクト変形方法。

【請求項 1 0】

複数の頂点により形状が定められる三次元オブジェクトを変形するビデオゲーム装置であって、

複数のジョイントにより構成され、且つ前記複数の頂点の各々が前記複数のジョイントのいずれか 1 つに対応付けられている前記三次元オブジェクトの仮想スケルトンにおける各前記ジョイントの回転角を、当該仮想スケルトンの動きが定義されたアニメーション・データに基づいてフレーム表示周期毎に取得する取得手段と、

前記取得手段により取得された各前記ジョイントの回転角と当該ジョイントに対応する前記頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、当該回転角に従って前記頂点を前記フレーム表示周期毎に移動する移動手段と

を含むことを特徴とするビデオゲーム装置。

【請求項 1 1】

複数の頂点により形状が定められる三次元オブジェクトを変形するビデオゲーム装置であって、

コンピュータと、

前記コンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体と、

を有し、

前記プログラムは、前記コンピュータに、

複数のジョイントにより構成され、且つ前記複数の頂点の各々が前記複数のジョイントのいずれか 1 つに対応付けられている前記三次元オブジェクトの仮想スケルトンにおける各前記ジョイントの回転角を、当該仮想スケルトンの動きが定義されたアニメーション・データに基づいてフレーム表示周期毎に取得する取得

処理と、

前記取得処理により取得された各前記ジョイントの回転角と当該ジョイントに対応する前記頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、当該回転角に従って前記頂点を前記フレーム表示周期毎に移動する移動処理と

を実行させるビデオゲーム装置。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】

本発明は、コンピュータ・グラフィックス（CG：Computer Graphics）に関し、より詳しくは、仮想三次元空間における三次元オブジェクトを変形する技術に関する。

【0002】

【従来の技術】

CGにおいて、複数の頂点により形状が定められている三次元オブジェクトの動作を表現するために、当該三次元オブジェクトを変形する技術としては、従来からスケルトン変形という技術が知られている。スケルトン変形では、三次元オブジェクトに対して仮想的に複数の関節（以下ジョイントと呼ぶ）と複数の骨（以下ボーンと呼ぶ）を設定し、ジョイントとボーンを階層的に接続した骨格情報（以下スケルトンと呼ぶ）を作成する。そして、スケルトンを変形することにより三次元オブジェクトを変形する。

【0003】

例えば、人型の三次元オブジェクトにおいて首の部分に首ジョイントを設ける。また、首ジョイントには頭部ボーンが接続され、頭部ボーンには人型の三次元オブジェクトの頭部が対応付けられているとする。この場合、首ジョイントが回転すれば、頭部ボーンも回転する。頭部ボーンに対応する人型の三次元オブジェクトの頭部は頭部ボーンの回転に応じて回転する。結果としてスケルトンを変形することにより、対応する三次元オブジェクトを変形することができる。

【0004】

但し、単純に、スケルトンを変形することにより、対応する三次元オブジェクトを変形させると、三次元オブジェクトの頂点が三次元オブジェクト内部にめり込んだり、一部のポリゴンが大きくなりすぎて、滑らかな三次元オブジェクトを表示できなくなる場合がある。

【 0 0 0 5 】

そこで、特開平 1 0 - 7 4 2 7 0 号等に示されるような技術が発明された。特開平 1 0 - 7 4 2 7 0 号公報には、子仮想スケルトンの方向情報 $\theta 1$ を変化させる場合、子仮想スケルトンに対応するポリゴンの頂点の位置情報を、変化前の頂点の位置情報と、影響度情報を用いずに方向情報 $\theta 1$ により求められた頂点の位置情報と、各頂点に設定された子仮想スケルトン及び親仮想スケルトンからの影響度情報とに基づいて求めるという技術が開示されている。

【 0 0 0 6 】

なお、特開平 1 0 - 7 4 2 7 0 号では、変化前の頂点の位置情報を用いるとしているが、子仮想スケルトンより上の階層の仮想スケルトンの移動が無ければ、変化前の頂点の位置情報を用いることができる。しかし、もしそうでなければ親仮想スケルトンまでの全ての移動を考慮した状態における頂点の位置情報を計算しなければならない。多くの場合、親仮想スケルトンまでの全ての移動を考慮した状態における頂点の位置情報、および影響度情報を用いずに方向情報 $\theta 1$ により求められた頂点の位置情報の 2 つの位置情報を頂点変換により求めなければならない。

【 0 0 0 7 】

すなわち、特開平 1 0 - 7 4 2 7 0 号公報の技術では、形状の滑らかさを要求しない場合に比べて、各頂点に対して頂点変換が 1 回ずつ多く行われる。そのため、三次元オブジェクトを表現する頂点数の増加に伴い、形状を滑らかに保つために必要な処理量も増大する。

【 0 0 0 8 】

【発明が解決しようとする課題】

一般に、三次元オブジェクトの形状の滑らかさを追求するほど、三次元オブジェクトの形状を定義する頂点数が増加する。そのため、三次元オブジェクトの頂

点数が増加した場合であっても、変形後の形状を滑らかにするために必要な処理の増加量は、極力低く抑えられる事が望まれている。

【 0 0 0 9 】

本発明の目的は、三次元オブジェクトの頂点数の増加に伴う処理負荷の増加量を極力少なくして、滑らかな形状を保った三次元オブジェクトの変形を行うことが可能な三次元オブジェクト変形技術を提供することである。

【 0 0 1 0 】

【課題を解決するための手段】

本発明の第 1 の態様に係る、複数の頂点により形状が定められる三次元オブジェクトを変形する、ビデオゲームにおける三次元オブジェクト変形方法は、複数のジョイントにより構成され、且つ複数の頂点の各々が複数のジョイントのいずれか 1 つに対応付けられている三次元オブジェクトの仮想スケルトンにおける各ジョイントの回転角を、当該仮想スケルトンの動きが定義されたアニメーション・データに基づいてフレーム表示周期毎に取得する第 1 ステップと、第 1 ステップで取得された各ジョイントの回転角と当該ジョイントに対応する頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、当該回転角に従って頂点をフレーム表示周期毎に移動する第 2 ステップとを含む。

【 0 0 1 1 】

頂点に予め定義された重みに従って当該頂点に対応するジョイントの回転角を変更することにより、当該頂点の回転角が求められる。頂点の回転角に従って頂点を移動することで、頂点変換は 1 度で済むようになる。

【 0 0 1 2 】

上で述べた第 2 ステップを、第 1 ステップで取得されたジョイントの回転角により定まる 1 つの回転軸に対する 1 つの回転角と当該ジョイントに対応する頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、1 つの回転軸に沿って、当該回転角だけ当該頂点を回転させて得られる位置に頂点をフレーム表示周期毎に移動するステップとする構成も可能である。頂点を最短距離にて移動させるものである。

【 0 0 1 3 】

また上で述べた第1ステップにおいて、互いに直交する3つの回転軸毎の各ジョイントの回転角を取得し、上で述べた第2ステップを、第1ステップで取得されたジョイントの3つの回転軸毎の回転角に基づき、一つの回転軸と一つの回転角とを計算する第3ステップと、第3ステップで計算された一つの回転角を、頂点に予め定義された重みに従って補間し、当該頂点の回転角を計算する第4ステップと、第4ステップで計算された回転角と一つの回転軸とに従って頂点をフレーム表示周期毎に移動させる第5ステップとを含むような構成とすることも可能である。

【0014】

また、上で述べた第4ステップを、一つの回転角を、頂点に予め定義された重みに従って球状線形補間し、当該頂点の回転角を計算するステップとすることも可能である。

【0015】

本発明の第1の態様において、1つのジョイントに対応する頂点は当該頂点に予め定義された重みによりソートされており、上で述べた第2ステップを、直前に移動された頂点に予め定義された重みと同じ重みが移動すべき頂点に定義されているか判断する第3ステップと、第3ステップにおいて直前に移動された頂点に予め定義された重みと同じ重みが移動すべき頂点に定義されていないと判断された場合には、移動すべき頂点に対応するジョイントの第1ステップで取得された回転角と当該移動すべき頂点に予め定義された重みとに基づいて当該移動すべき頂点の回転角を算出し、当該回転角に関する情報を格納する第4ステップと、第4ステップにおいて格納された回転角に従って、移動すべき頂点を移動させる第5ステップと、第3ステップにおいて直前に移動された頂点に予め定義された重みと同じ重みが移動すべき頂点に定義されていると判断された場合には、第5ステップで格納された回転角に従って前記移動すべき頂点を移動させる第6ステップとを含むような構成にすることも可能である。

【0016】

また、上で述べた第2ステップを、同一の前記ジョイントに対応した頂点のうち既に移動された頂点に予め定義されている重みと同一の重みが移動すべき頂点

に定義されているか判断する第3ステップと、第3ステップにおいて同一のジョイントに対応した頂点のうち既に移動された頂点に予め定義されている重みと同一の重みが移動すべき頂点に定義されていないと判断された場合には、移動すべき頂点に対応するジョイントの第1ステップで取得された回転角と当該移動すべき頂点に予め定義された重みとに従って当該頂点の回転角を算出し、当該回転角に関する情報を当該頂点の重みと関係付けて格納する第4ステップと、第4ステップにおいて算出された回転角に従って、移動すべき頂点を移動させる第5ステップと、第3ステップにおいて同一のジョイントに対応した頂点のうち既に移動された頂点に予め定義されている重みと同一の重みが移動すべき頂点に定義されていると判断された場合には、移動すべき頂点に予め定義された重みに関係付けて第4ステップで格納された回転角を取得し、当該回転角に従って、当該移動すべき頂点を移動させる第6ステップとを含むような構成も可能である。

【0017】

本発明の第1の態様に係るビデオゲームにおける三次元オブジェクト変形方法をコンピュータに実行させるプログラムを作成することは可能である。その際、第1の態様に対する上記のような変形は、当該プログラムに対しても応用可能である。本発明に係るプログラムは、例えばCD-ROM (Compact Disc - Read Only Memory)、DVD (Digital Versatile Disc)、フロッピーディスク、メモリカートリッジ、メモリ、ハードディスクなどの記録媒体又は記憶装置に格納される。記録媒体又は記憶装置に格納されるプログラムをコンピュータに読み込ませることで以下で述べるビデオゲーム装置を実現できる。また、記録媒体によって本発明に係るプログラムをソフトウェア製品として装置と独立して容易に配布、販売することができるようになる。さらに、コンピュータなどのハードウェアを用いて本発明に係るプログラムを実行することにより、コンピュータ等のハードウェアで本発明のグラフィックス技術が容易に実施できるようになる。

【0018】

本発明の第2の態様に係る、複数の頂点により形状が定められる三次元オブジェクトを変形するビデオゲーム装置は、複数のジョイントにより構成され、且つ複数の頂点の各々が複数のジョイントのいずれか1つに対応付けられている三次

元オブジェクトの仮想スケルトンにおける各ジョイントの回転角を、当該仮想スケルトンの動きが定義されたアニメーション・データに基づいてフレーム表示周期毎に取得する取得手段と、取得手段により取得された各ジョイントの回転角と当該ジョイントに対応する頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、当該回転角に従って頂点をフレーム表示周期毎に移動する移動手段とを含む。

【 0 0 1 9 】

さらに本発明の第 3 の態様に係るビデオゲーム装置は、コンピュータと、コンピュータに実行させるプログラムを記録したコンピュータ読み取り可能な記録媒体とを有する。そして、当該プログラムは、コンピュータに、複数のジョイントにより構成され、且つ複数の頂点の各々が複数のジョイントのいずれか 1 つに対応付けられている三次元オブジェクトの仮想スケルトンにおける各ジョイントの回転角を、当該仮想スケルトンの動きが定義されたアニメーション・データに基づいてフレーム表示周期毎に取得する取得処理と、取得処理により取得された各ジョイントの回転角と当該ジョイントに対応する頂点に予め定義された重みとに基づいて当該頂点の回転角を算出し、当該回転角に従って頂点をフレーム表示周期毎に移動する移動処理とを実行させる。

【 0 0 2 0 】

【発明の実施の形態】

本発明をコンピュータ・プログラムにより実施する場合において当該コンピュータ・プログラムを実行する家庭用ゲーム機 1 0 1 の一例を図 1 に示す。家庭用ゲーム機 1 0 1 は、例えば内部バス 1 1 9 に接続された CPU (Central Processing Unit) 1 0 3、ROM (Read Only Memory) 1 0 4、RAM (Random Access Memory) 1 0 5、HDD (Hard Disk Drive) 1 0 7、サウンド処理部 1 0 9、グラフィックス処理部 1 1 1、CD-ROMドライブ 1 1 3、通信インターフェース 1 1 5、及びインターフェース部 1 1 7 を備える。グラフィックス処理部 1 1 1 は、フレームバッファ 1 1 2 を備える。

【 0 0 2 1 】

家庭用ゲーム機 1 0 1 のサウンド処理部 1 0 9 及びグラフィックス処理部 1 1

1は表示画面120を有するTVセット121に接続されている。また、CD-ROMドライブ113には着脱可能なCD-ROM131が装着される。CD-ROM131には、本発明にかかるゲームプログラム133及びデータ135が記録されている。通信インターフェース115はネットワーク151と通信媒体141を介して接続される。インターフェース部117には、操作ボタンを備えたキーパッド161及びメモリカード171が接続される。

【0022】

CPU103は、ROM104に格納されたプログラム、及びCD-ROM131上に記録されたゲームプログラム133を実行し、家庭用ゲーム機101の制御を行う。RAM105は、CPU103のワークエリアである。HDD107は、例えばCD-ROM131上に記録されたゲームプログラム133及びデータ135を保管するための記憶領域である。メモリカード171は、ゲームプログラム133により参照されるデータを保存するための記憶領域である。サウンド処理部109は、CPU103により実行されているプログラムがサウンド出力を行うよう指示している場合に、その指示を解釈して、TVセット121にサウンド信号を出力する。

【0023】

グラフィックス処理部111は、CPU103から出力される描画命令に従って、画像データを生成してフレームバッファ112に書き込む。そして、書き込んだ画像データを表示画面120に表示するための信号をTVセット121に出力する。CD-ROMドライブ113は、CD-ROM131上のゲームプログラム133及びデータ135を読み出す。通信インターフェース115は、通信媒体141を介してネットワーク151に接続され、他のコンピュータ等との間で行われるデータ通信の入出力制御を行う。インターフェース部117は、キーパッド161からの入力をRAM105に出力し、CPU103がキーパッド161からの入力を解釈して演算処理を実施する。

【0024】

本発明に係るゲームプログラム133及びデータ135は最初例えばCD-ROM131に記録されている。そして、ゲームプログラム133及びデータ13

5は実行時にCD-ROMドライブ113により読み出されて、RAM105にロードされる。RAM105にロードされるデータ135としては、図1に示したように、頂点座標リスト200、法線ベクトル・リスト300、三角形ポリゴン・リスト400、頂点グループ・リスト500、クラスタ・データ600、及びアニメーション・データ700である。また、本発明に係るプログラムの処理途中で使用されるデータとして、マトリックス・バッファ800、頂点バッファ850、法線ベクトル・バッファ870、マトリックス・キャッシュ900、及びフレーム・カウンタ1100とがある。これらについては以下に詳しく説明する。なお、CD-ROM131に記録されている本発明に係るゲームプログラム133及びデータ135を、予めCD-ROMドライブ113により読み出し、HDD107に格納しておくようにしてもよい。HDD107に本発明に係るゲームプログラム133及びデータ135が格納されている場合には、HDD107からRAM105にゲームプログラム133及びデータ135がロードされる。

【0025】

CPU103はRAM105にロードされた、本発明に係るゲームプログラム133及びデータ135を処理し、描画命令をグラフィックス処理部111に出力する。なお、中間的なデータはRAM105に記憶される。グラフィックス処理部111はCPU103からの描画命令に従って処理を行い、画像データをフレームバッファ112に書き込み、表示画面120に表示するための信号をTVセット121に出力する。

【0026】

以上のような家庭用ゲーム機101において実行される本発明のプログラムのアルゴリズム及び使用されるデータについて以下で詳しく述べる。

【0027】

図2乃至図9に本発明で使用するデータの説明を行う。

【0028】

三次元オブジェクトが三角形ポリゴンにより構成される場合において、三次元オブジェクトの頂点の頂点座標リスト200を図2に示す。頂点座標リスト200

0 は、頂点座標インデックス (index) の欄 2 1 0 と、座標 (x, y, z) の欄 2 2 0 とを含む。

【 0 0 2 9 】

図 2 において、頂点インデックス $vtx0$ の頂点の座標は (0, 0, 0) である。頂点インデックス $vtx1$ の頂点の座標は (0, 0, 100) である。頂点インデックス $vtx2$ の頂点の座標は (0, 100, 0) である。頂点インデックス $vtx3$ の頂点の座標は (0, 100, 100) である。頂点インデックス $vtx4$ の頂点の座標は (100, 0, 0) である。頂点座標リスト 200 における頂点座標は、仮想スケルトンにおいて対応するジョイントからの相対的な位置座標である。

【 0 0 3 0 】

図 3 に法線ベクトル・リスト 300 を示す。法線ベクトル・リスト 300 は、三角形ポリゴンの各頂点における法線ベクトルを定義するためのリストである。法線ベクトル・リスト 300 は、法線ベクトル・インデックス (index) の欄 3 1 0 と法線ベクトル (x, y, z) の欄 3 2 0 とを含む。

【 0 0 3 1 】

図 3 において、法線ベクトル・インデックス $nm10$ の法線ベクトルは (1, 0, 0) である。法線ベクトル・インデックス $nm11$ の法線ベクトルは (-1, 0, 0) である。法線ベクトル・インデックス $nm12$ の法線ベクトルは (0, 0, -1) である。法線ベクトル・インデックス $nm13$ の法線ベクトルは (-1, -1, 0) である。法線ベクトル・インデックス $nm14$ の法線ベクトルは (0, -1, 0) である。

【 0 0 3 2 】

図 4 に三角形ポリゴン・リスト 400 を示す。三角形ポリゴン・リスト 400 は、各三角形ポリゴンを構成する頂点 0 乃至 2 の頂点インデックス及び法線ベクトル・インデックスを格納する。三角形ポリゴン・リスト 400 は、三角形ポリゴン・インデックス (index) の欄 4 1 0 と、頂点 0 を構成する頂点の頂点インデックス及び法線ベクトルの法線ベクトル・インデックスのための頂点 0 の欄 4 2 0 と、頂点 1 を構成する頂点の頂点インデックス及び法線ベクトルの法線ベク

トル・インデックスのための頂点1の欄430と、頂点2を構成する頂点の頂点インデックス及び法線ベクトルの法線ベクトル・インデックスのための頂点2の欄440とを含む。

【0033】

図4において、三角形ポリゴン・インデックスp1g0の三角形ポリゴンの頂点0は、頂点インデックスvtx0の頂点及び法線ベクトル・インデックスnm10の法線ベクトルにて構成される。三角形ポリゴン・インデックスp1g0の三角形ポリゴンの頂点1は、頂点インデックスvtx1の頂点及び法線ベクトル・インデックスnm12の法線ベクトルにて構成される。三角形ポリゴン・インデックスp1g0の三角形ポリゴンの頂点2は、頂点インデックスvtx2の頂点及び法線ベクトル・インデックスnm13の法線ベクトルにて構成される。

【0034】

三角形ポリゴン・インデックスp1g1の三角形ポリゴンの頂点0は、頂点インデックスvtx0の頂点及び法線ベクトル・インデックスnm11の法線ベクトルにて構成される。三角形ポリゴン・インデックスp1g1の三角形ポリゴンの頂点1は、頂点インデックスvtx2の頂点及び法線ベクトル・インデックスnm13の法線ベクトルにて構成される。三角形ポリゴン・インデックスp1g0の三角形ポリゴンの頂点2は、頂点インデックスvtx3の頂点及び法線ベクトル・インデックスnm14の法線ベクトルにて構成される。

【0035】

頂点座標リスト200と法線ベクトル・リスト300とを設け、頂点インデックス及び法線ベクトル・インデックスとを格納する三角形ポリゴン・リスト400を別途用意することにより、同じ頂点座標のデータや同じ法線ベクトルのデータを三角形ポリゴン・リスト400において共用することができる。また、同じ頂点にも法線ベクトルを別途定義することが容易になる。但し、1つの法線は1つの頂点にしか対応させることができない。

【0036】

三次元オブジェクトの形状を定める頂点はグループ分けされている。図5に頂点グループ・リスト500を示す。頂点グループ・リスト500には、頂点グル

ープ数と、頂点グループ毎に、頂点グループに属する頂点（メンバ）の数、その頂点グループに属する頂点の各頂点インデックス、各頂点に予め定義された重み、各頂点に定義される法線数、及び各頂点の法線インデックスとが格納される。よって、頂点グループ・リスト 5 0 0 には、グループ数の欄 5 1 0 と、頂点数の欄 5 2 0 乃至 5 2 4 と、頂点インデックスの欄 5 3 0 乃至 5 5 4 と、頂点の重みの欄 5 4 0 乃至 5 6 4 と、法線数の欄 5 7 0 乃至 5 8 4 と、法線ベクトル・インデックスの欄 5 8 6 乃至 5 9 9 とが設けられている。頂点数の欄が出てくると新たな頂点グループについてのデータが始まる。頂点グループ番号は頂点グループ・リスト 5 0 0 の先頭のデータ（図中上に位置するデータ）から順に割り付けられる。

【 0 0 3 7 】

図 5 において、グループ数の欄 5 1 0 はグループ数が 3 であることを示している。頂点グループ 0 の頂点数の欄 5 2 0 は頂点グループ 0 の頂点数が 5 であることを示している。頂点グループ 0 に属する頂点の頂点インデックスの欄 5 3 0 には頂点インデックス vtx_0 が格納されている。頂点インデックス vtx_0 に対する重みの欄 5 4 0 には重み 1. 0 が格納されている。頂点インデックス vtx_0 に対する法線数の欄 5 7 0 には 2 が格納されている。頂点インデックス vtx_0 に対する法線ベクトル・インデックスの欄 5 8 6 には、法線ベクトル・インデックス nml_0 及び nml_1 が格納されている。

【 0 0 3 8 】

頂点グループ 0 に属する頂点の頂点インデックスの欄 5 3 2 には頂点インデックス vtx_5 が格納されている。頂点インデックス vtx_5 に対する重みの欄 5 4 2 には重み 1. 0 が格納されている。頂点インデックス vtx_5 に対する法線数の欄 5 7 2 には 1 が格納されている。頂点インデックス vtx_5 に対する法線ベクトル・インデックスの欄 5 8 8 には、法線ベクトル・インデックス nml_1 0 が格納されている。

【 0 0 3 9 】

頂点グループ 0 に属する頂点の頂点インデックスの欄 5 3 4 には頂点インデックス vtx_1 が格納されている。頂点インデックス vtx_1 に対する重みの欄 5

44には重み0.7が格納されている。頂点インデックス $vtx1$ に対する法線数の欄574には1が格納されている。頂点インデックス $vtx1$ に対する法線ベクトル・インデックスの欄590には、法線ベクトル・インデックス $nm12$ が格納されている。

【0040】

頂点グループ0に属する頂点の頂点インデックスの欄536には頂点インデックス $vtx2$ が格納されている。頂点インデックス $vtx2$ に対する重みの欄546には重み0.6が格納されている。頂点インデックス $vtx2$ に対する法線数の欄576には2が格納されている。頂点インデックス $vtx2$ に対する法線ベクトル・インデックスの欄592には、法線ベクトル・インデックス $nm13$ 及び $nm11$ が格納されている。

【0041】

頂点グループ0に属する頂点の頂点インデックスの欄538には頂点インデックス $vtx3$ が格納されている。頂点インデックス $vtx3$ に対する重みの欄548には重み0.5が格納されている。頂点インデックス $vtx3$ に対する法線数の欄578には1が格納されている。頂点インデックス $vtx3$ に対する法線ベクトル・インデックスの欄594には、法線ベクトル・インデックス $nm14$ が格納されている。

【0042】

頂点グループ1の頂点数の欄522は頂点グループ1の頂点数が2であることを示している。頂点グループ1に属する頂点の頂点インデックスの欄550には頂点インデックス $vtx4$ が格納されている。頂点インデックス $vtx4$ に対する重みの欄560には重み0.3が格納されている。頂点インデックス $vtx4$ に対する法線数の欄580には1が格納されている。頂点インデックス $vtx4$ に対する法線ベクトル・インデックスの欄596には、法線ベクトル・インデックス $nm15$ が格納されている。

【0043】

頂点グループ1に属する頂点の頂点インデックスの欄552には頂点インデックス $vtx6$ が格納されている。頂点インデックス $vtx6$ に対する重みの欄5

62には重み0.2が格納されている。頂点インデックス $vtx6$ に対する法線数の欄582には3が格納されている。頂点インデックス $vtx6$ に対する法線ベクトル・インデックスの欄598には、法線ベクトル・インデックス $nm16$ 、 $nm18$ 及び $nm19$ が格納されている。

【0044】

頂点グループ2の頂点数の欄522は頂点グループ2の頂点数が1であることを示している。頂点グループ2に属する頂点の頂点インデックスの欄554には頂点インデックス $vtx7$ が格納されている。頂点インデックス $vtx7$ に対する重みの欄564には重み0.5が格納されている。頂点インデックス $vtx7$ に対する法線数の欄584には1が格納されている。頂点インデックス $vtx7$ に対する法線ベクトル・インデックスの欄599には、法線ベクトル・インデックス $nm17$ が格納されている。

【0045】

本実施の形態では、重みは0以上1以下の実数である。また、頂点は一つの頂点グループに属している。

【0046】

図6には、人型の三次元オブジェクト10を前方から見たときの仮想スケルトンの例を示している。人型の三次元オブジェクト10に対応する仮想スケルトン20には、ルート・ジョイントJ2と、ジョイントJ3乃至J15と、ルート・ボーンB2と、ボーンB3乃至B15とが含まれる。

【0047】

ボーンB3は、ジョイントJ4とジョイントJ5との間に観念される。ボーンB5は、ジョイントJ5とジョイントJ6との間に観念される。ボーンB8は、ジョイントJ6に接続されるように観念される。ルート・ボーンB2は、ジョイントJ3とジョイントJ7との間に観念される。ボーンB6は、ジョイント7とジョイントJ8との間に観念される。ボーンB7は、ジョイントJ8に接続されるように観念される。

【0048】

ボーンB4は、ルート・ジョイントJ2とジョイントJ9との間に観念される

。ボーンB9は、ジョイントJ9に接続されるように観念される。ボーンB11は、ジョイントJ11とジョイントJ14との間に観念される。ボーンB14は、ジョイントJ14とジョイントJ15との間に観念される。ボーンB15は、ジョイントJ15に接続されるように観念される。ボーンB10は、ジョイントJ10とジョイントJ12との間に観念される。ボーンB12は、ジョイントJ12とジョイントJ13の間に観念される。ボーンB13は、ジョイントJ13に接続されるように観念される。

【0049】

仮想スケルトン20は文字通り仮想的なスケルトンであり、表示画面120には表示されない。また、人型の三次元オブジェクト10であっても、実際の人間には存在しないジョイントが設けられ、またボーンも観念されている。

【0050】

なお、図6にはボーンは示されているが、仮想スケルトン20を容易に理解できるようにするために示されているだけであって、本実施の形態では仮想スケルトン20においてボーンのパータは無い。よって、上においてボーンについては「観念される」としている。

【0051】

各ジョイントは階層構造を成している。図6の例では、図7に示すような階層構造である。ルート・ジョイントJ2は、腰の付け根に存在するジョイントであり、最上位階層に属する。このルート・ジョイントJ2の子ジョイントは、右足の付け根のジョイントJ3、左足の付け根のジョイントJ4、首の付け根のジョイントJ9、左腕の付け根のジョイントJ10、及び右腕の付け根のジョイントJ11である。ジョイントJ3は三次元オブジェクト10の右足の付け根のジョイントである。よって、ジョイントJ3の子ジョイントは、三次元オブジェクト10の右膝のジョイントであるジョイントJ7である。ジョイントJ7の子ジョイントは、三次元オブジェクト10の右足首のジョイントであるジョイントJ8である。

【0052】

ジョイントJ4は三次元オブジェクト10の左足付け根のジョイントである。

よって、ジョイント J 4 の子ジョイントは、三次元オブジェクト 1 0 の左膝のジョイントであるジョイント J 5 である。ジョイント J 5 の子ジョイントは、三次元オブジェクト 1 0 の左足首のジョイントであるジョイント J 6 である。

【 0 0 5 3 】

ジョイント J 1 0 は、三次元オブジェクト 1 0 の左腕の付け根のジョイントである。よって、ジョイント J 1 0 の子ジョイントは、三次元オブジェクト 1 0 の左肘のジョイントであるジョイント J 1 2 である。ジョイント J 1 2 の子ジョイントは、三次元オブジェクトの左手首のジョイントであるジョイント J 1 3 である。

【 0 0 5 4 】

ジョイント J 1 1 は、三次元オブジェクト 1 0 の右腕の付け根のジョイントである。よって、ジョイント J 1 1 の子ジョイントは、三次元オブジェクト 1 0 の右肘のジョイントであるジョイント J 1 4 である。ジョイント J 1 4 の子ジョイントは、右手首のジョイントであるジョイント J 1 5 である。

【 0 0 5 5 】

図 7 に示したように、ルート・ジョイント J 2 を除く各ジョイントには、必ず 1 つのジョイントが親ジョイントとして対応付けられている。なお子ジョイントはいくつであってもよいし、無くてよい。

【 0 0 5 6 】

各ジョイントの持つ変化量に従って当該ジョイントに対応付けられた頂点は移動する。同時に、当該ジョイントの子ジョイントもすべて移動する。例えばルート・ジョイント J 2 が平行移動すれば、ルート・ジョイント J 2 の下位のジョイント全て、すなわち仮想スケルトン 2 0 全体が平行移動する。これにより対応する三次元オブジェクト 1 0 全体が平行移動する。なお、ルート・ジョイント J 2 の移動は全てのジョイントに影響を及ぼすので、ルート・ジョイント J 2 は三次元オブジェクト 1 0 の配置等にも用いられる。

【 0 0 5 7 】

本実施の形態では上で述べたような仮想スケルトンを使用する。そして、各仮想スケルトンのジョイントについては、図 8 に示すようなジョイント・データ 6

00が用意される。ジョイント・データ600には、ジョイントのデータと、ジョイントに対応付けられる頂点グループのデータとがジョイントごとに格納される。ジョイントのデータには、ジョイントの階層構造に関するデータも含まれる。

【0058】

ジョイント・データ600において、ルート・インデックス(index)の欄610には、ルート・ジョイントのジョイント・インデックスが格納される。図8の例ではrootである。ルート・ジョイントの基準平行移動量(x, y, z)の欄612には、仮想スケルトンの基準姿勢におけるルート・ジョイントの基準となる位置が格納される。図8の例では(0, 0, 0)である。ルート・ジョイントの基準回転角(Rx, Ry, Rz)の欄614には、ルート・ジョイントの基準となる回転角が格納される。図8の例では(0, 0, 90)である。ジョイント数の欄616には、ルート・ジョイント以下の階層に属するジョイントの数が格納される。図8の例では4である。

【0059】

ジョイント・インデックス(index)の欄620には、ルート・ジョイントより下位の階層に属するジョイントのジョイント・インデックスjoint0が格納される。親インデックス(index)の欄622には、ジョイントjoint0の親ジョイントのジョイント・インデックスrootが格納される。基準平行移動量の欄624には、仮想スケルトンの基本姿勢におけるジョイントjoint0の親ジョイントrootからの基準となる相対座標(90, 20, 0)が格納される。基準回転角の欄626には、仮想スケルトンの基本姿勢におけるジョイントjoint0の親ジョイントからの基準となる相対的な回転角(0, 0, 135)が格納される。

【0060】

頂点グループ数の欄628には、ジョイントjoint0に対応する頂点グループの数が格納される。図8の例では2である。頂点グループ番号の欄630及び632には、具体的にジョイントjoint0に対応する頂点グループ・リスト500における頂点グループ番号が格納される。図8の例では、頂点グループ0と、頂点グループ2である。

【 0 0 6 1 】

ジョイント・インデックスの欄 6 4 0 には、ルート・ジョイント root 及びジョイント joint0 以外の仮想スケルトンに属するジョイントのジョイント・インデックス (index) が格納される。図 8 の例では、ジョイント・インデックス joint1 である。ジョイント joint1 の親ジョイントのインデックスを格納する親インデックス (index) の欄 6 4 2 には、ジョイント・インデックス joint0 が格納される。

【 0 0 6 2 】

基準平行移動量の欄 6 4 4 には、仮想スケルトンの基本姿勢におけるジョイント joint1 の親ジョイント joint0 からの基準となる相対座標 (4 0, 0, 0) が格納される。基準回転角の欄 6 4 6 には、仮想スケルトンの基本姿勢におけるジョイント joint1 の親ジョイント joint0 からの基準となる相対的な回転角 (0, 0, 4 5) が格納される。

【 0 0 6 3 】

頂点グループ数の欄 6 4 8 には、ジョイント joint1 に対応する頂点グループの数が格納される。図 8 の例では 1 である。頂点グループ番号の欄 6 5 0 には、具体的にジョイント joint1 に対応する頂点グループ・リスト 5 0 0 における頂点グループ番号が格納される。図 8 の例では、頂点グループ 1 である。

【 0 0 6 4 】

仮想スケルトンの各ジョイントに対して変形を行うための操作は、図 9 に示されたアニメーション・データ 7 0 0 に従って行われる。すなわち、クラスタ数及びフレーム数を定義するアニメーション・データ 7 0 0 A (図 9 (a)) と、変形開始から終了までの各フレーム表示周期毎に移動する各クラスタの回転角を定義するアニメーション・データ 7 0 0 B (図 9 (b)) とが含まれる。なお、アニメーション・データ 7 0 0 B においてジョイントは、ルート・ジョイントから順に最下位の階層へ並べられる。

【 0 0 6 5 】

図 9 (a) においては、移動するジョイントの数がジョイント数の欄 7 1 0 に格納される。図 9 (a) の例では 2 である。そして、変形開始から終了までのタ

イム・フレーム数は、フレーム数の欄 7 2 0 に格納される。図 9 (a) の例では 1 0 である。以下、移動するジョイントの数 2 に対応して 2 行毎にフレーム表示周期のジョイントの回転角がアニメーション・データ 7 0 0 B (図 9 (b)) に規定される。ジョイント・インデックス (index) の欄 7 3 には、移動するジョイントのインデックスが格納される。回転角 (R_x , R_y , R_z) の欄 7 4 0 には、基準姿勢からの回転角が格納される。

【 0 0 6 6 】

フレーム 0 において、移動するジョイント joint0 の回転角は (0, 0, 0) である。また移動するジョイント joint1 の回転角も (0, 0, 0) である。これは、基本姿勢のままであるということを意味する。次に、フレーム 1 において、ジョイント joint0 の回転角は (1 0, 0, 1 3 5) である。またジョイント joint1 の回転角は (-1 0, 5, 4 5) である。このようなデータがフレーム 9 まで続く。フレーム 9 において、ジョイント joint0 の回転角は (9 0, 0, 1 5 0) である。ジョイント joint1 の回転角は (-9 0, 4 5, 4 5) である。

【 0 0 6 7 】

なお、本実施の形態では、回転角は R_x , R_y , R_z の 3 つのパラメータを有する。 R_x は、ジョイントを通る x 軸を中心に R_x だけ回転することを意味する。 R_y は、ジョイントを通る y 軸を中心に R_y だけ回転することを意味する。 R_z は、ジョイントを通る z 軸を中心に R_z だけ回転することを意味する。ここでは $x y z$ の順番に回転することとするが、回転の順番は任意である。但し、回転の順番が異なれば回転後の位置も異なるので、データ作成時と仮想スケルトン変形時において回転の順序は統一されている必要がある。

【 0 0 6 8 】

以上述べたようなデータを本実施の形態では用いる。本実施の形態におけるプログラムの処理について説明する前に、動作の原理について図 1 0 を用いて説明しておく。

【 0 0 6 9 】

図 1 0 (a) は、二次元の仮想スケルトンを変形する前の状態を示している。仮想スケルトンはジョイント A 及び B と、ボーン B 1 及び B 2 により構成される

。ポリゴンの頂点V1乃至V4はジョイントAに対応している。ポリゴンの頂点V5乃至V8はジョイントBに対応している。頂点V1乃至V4の重みは皆100%すなわち1.0である。頂点V5及びV6の重みは50%すなわち0.5である。頂点V7及びV8の重みは100%すなわち1.0である。図10(a)において頂点記号の下に括弧内は対応するジョイントの記号と重みを示している。

【0070】

図10(b)は、仮想スケルトンのジョイントBを60度回転した状態を示している。但し、図10(a)のように頂点V5及びV6の重みを50%ではなく100%とした場合である。頂点V6aは三次元オブジェクト内部にめり込んでしまうことが分かる。これでは、滑らかな三次元オブジェクトの表示を行うことができない。

【0071】

一方、図10(a)において頂点V5及びV6のジョイントAに対する影響度を50%、ジョイントBに対する影響度を50%として、例えば特開平10-74270号公報に述べられている方法で変形した場合を図10(c)に示す。図10(b)において頂点V5及びV6は、変形前の図10(a)と同じである。V5a及びV6aは、ジョイントBからの影響度を100%とした場合の頂点V5及びV6の変形後の頂点位置を示している。図10(c)に示したV5a及びV6aと同じである。図10(c)における点線は図10(b)に示された頂点を結ぶ線である。

【0072】

特開平10-74270号公報記載の技術では、V5とV5aの位置座標をジョイントA及びBへの影響度により補間する。V6とV6aの位置座標もジョイントA及びBへの影響度により補間する。補間により求められた頂点位置V5b及びV6bが、頂点V5及びV6の変形後の位置となる。従来技術の説明の欄においても述べたが、ジョイントBの親ジョイントであるジョイントAが移動しなければ上のような処理を実施することになるが、実際にはジョイントAが移動する可能性もあるので、V5及びV6の位置を得るため、及びV5a及びV6aの

位置を得るために、2度頂点変換を実施する必要がある。

【0073】

本発明においては、図10(a)に示したように、頂点V5及びV6には対応する1つのジョイントと1つの重みしか定義されない。図10(a)の例では頂点V5及びV6の重みは50%であった。本発明をしごく簡単にしてしまえば、ジョイントBを60度回転する場合には、頂点V5及びV6については $60(度) \times 0.5(重み) = 30度$ だけジョイントBを回転させた場合における位置を計算し、当該位置を頂点V5及びV6の変形後の頂点位置とするものである。

【0074】

図10(d)において点線で示されたボーンB2aが回転を30度にした場合を示している。ボーンB2を30度回転させた場合における頂点V5及びV6の移動先は、V5b及びV6bである。頂点V7及びV8はジョイントBに対して重み100%であるから、ボーンB2のように60度回転した場合に対応する位置に配置される。このようにすればV6が三次元オブジェクト内部にめり込むことなく、且つ変形後の三次元オブジェクトは滑らかである。

【0075】

なお、図10では回転の方向を1つの軸を中心にしたものに限定していたため、簡単に重みに対応する回転角が求められた。しかし、三次元空間において回転を処理する場合には、問題が生じ得るので注意が必要となる。

【0076】

例えば、x軸を中心に0度、y軸を中心に0度、z軸を中心に90度の順番で回転させる変換と、x軸を中心に180度、y軸を中心に180度、z軸を中心に-90度の順番で回転させる変換とは全く同じ変換となる。回転変換マトリックスも同じになる。

【0077】

しかし、重み50%の頂点のために、各軸の角度を重みに応じて単純に変更すると、好ましくない結果が現れる。すなわち、前者はx軸を中心に0度、y軸を中心に0度、z軸を中心に45度の順番で回転させる変換となる。後者はx軸を中心に90度、y軸を中心に90度、z軸を中心に-45度の順番で回転させる

変換となる。変更前では同じ変換であったのにもかかわらず、変更後では全く異なる変換となる。

【0078】

よって本実施の形態では、重みの反映をクォータニオン及び球状線形補間を用いて行う。以下、クォータニオン及び球状線形補間について簡単に説明しておく。

【0079】

クォータニオンは4元数とも呼ばれ、ベクトルとスカラー値の組み合わせで表現される。例えば、クォータニオン q は $q = a + b i + c j + d k$ である。 a , b , c 及び d はスカラー値で i , j 及び k は単位ベクトルである。 $q = (b, c, d, a) = q(V, a)$ (ベクトル $V = (b, c, d)$) といったような表記もされる。クォータニオンは回転軸を定めた回転を表現することができる。

【0080】

i を x 軸、 j を y 軸、 k を z 軸とする。 x 軸を中心に R_x 回転させる場合のクォータニオンは、 $q_x = (\sin(R_x/2), 0, 0, \cos(R_x/2))$ である。 y 軸を中心に R_y 回転させる場合のクォータニオンは、 $q_y = (0, \sin(R_y/2), 0, \cos(R_y/2))$ である。 z 軸を中心に R_z 回転させる場合のクォータニオンは、 $q_z = (0, 0, \sin(R_z/2), \cos(R_z/2))$ である。 x 軸 y 軸 z 軸の順番で回転させるクォータニオンは、 $q_{rot} = q_x * q_y * q_z$ で求められる (クォータニオンの乗算を示す記号を $*$ とする)。 q_{rot} は x 軸、 y 軸、 z 軸の順番での回転結果と同じ回転結果を1回の回転で得るための回転軸と回転角を表す。但し、クォータニオン同士の乗算は以下のようなになる。但し、「 \times 」は外積であり、「 \cdot 」は内積を表している。

$$\begin{aligned} q_1 * q_2 &= (V_1, w_1) * (V_2, w_2) \\ &= (V_2 * w_1 + V_1 * w_2 + V_1 \times V_2, w_1 * w_2 - V_1 \cdot V_2) \quad (1) \end{aligned}$$

【0081】

球状線形補間は図11のような状況を考えれば理解し易い。図11においてA、P及びBは円弧状に位置している。ベクトルAとベクトルBの間の角度は θ_2 であり、ベクトルAとベクトルPの間の角度は θ_1 である場合、ベクトルPを求

める。 $P = \alpha A + \beta B$ (α 及び β は実数) と書くことができ、 $|P| = 1$ 、 $A \cdot B = \cos \theta_2$ 、 $A \cdot P = \cos \theta_1$ であるから以下の式で求められる。

$$P = (A \sin(\theta_2 - \theta_1) + B \sin \theta_1) / \sin \theta_2. \quad (2)$$

【0082】

なお、 $\theta_1 = \theta_2 u$ (u は $[0, 1]$) とすると以下のようなになる。

$$P = (A \sin(1 - u) \theta_2 + B \sin(\theta_2 u)) / \sin \theta_2 \quad (3)$$

【0083】

これと同様のことをクォータニオンにおいても行えばよい。クォータニオン $q_0 = (V_0, w_0)$ と $q_1 = (V_1, w_1)$ を t ($[0, 1]$ に属する) で球状線形補間したクォータニオンは以下のようなになる。

$$q = (q_0 * \sin((1-t)\omega) + q_1 * \sin(t*\omega)) / \sin \omega \quad (4)$$

$$\omega = \arccos(w_0 * w_1 + V_0 \cdot V_1)$$

なお、 $(t*\omega)$ は、 t と ω のスカラー積を示す。また、「 \arccos 」はアークコサインを示す記号である。

【0084】

本実施の形態において図10(d)で示した頂点 V_5 及び V_6 の移動先 $V_5 b$ 及び $V_6 b$ を求めるには、単位クォータニオン $q_i = (0, 0, 0, 1)$ とジョイント B に対して行われる回転に対応するクォータニオンとの球状線形補間を行えば良い。但し、簡略化して、 $q_2 = q_i$ として、 $q = (1 - t) q_1 + q_i * t$ にて計算を行うことも可能である。

【0085】

以上の計算にて求められたクォータニオン $q = (x, y, z, w)$ に基づく回転変換マトリックスは、以下のようなになる。

【0086】

【数 1】

$$R = \begin{bmatrix} 1-2*(y*y+z*z) & 2*(x*y-w*z) & 2*(x*z+w*y) & 0 \\ 2*(x*y+w*z) & 1-2*(x*x+z*z) & 2*(y*z-w*x) & 0 \\ 2*(x*z-w*y) & 2*(y*z+w*x) & 1-2*(x*x+y*y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5)$$

式 (5) における「*」はスカラー積を示す記号である。

クォータニオン及び球状線形補間の詳細については、Advanced Animation and Rendering Techniques Theory and Practice (Alan Watt and Mark Watt, ADDISON-WESLEY ISBN 0-201-54412-1) を参照のこと。

【0087】

クォータニオン及び球状線形補間を使用して、頂点変換用マトリックスを生成することは、あたかも (a) ジョイントの回転角 (R_x , R_y , R_z) から定まる 1 つの回転軸とこの回転軸に対応する回転角とを求め、(b) さらに回転角をジョイントに対応する頂点に予め定義された重みに従い変更し、(c) 変更された回転角だけ対応する回転軸を中心に回転させる処理を行ったようなものである。結果として同じような回転を得る方法は多数あるが、クォータニオン及び球状線形補間を使用すると、実施すべき回転を最短距離で実施することになる。

【0088】

以上のような前提を元にして本実施の形態にかかる処理フローを説明する。

【0089】

CPU103 は、起動時に、ROM104 等に記憶されているオペレーティングシステムに基づき、CD-ROMドライブ113を介してCD-ROM131から画像処理やゲームの実行に必要なゲームプログラム133やデータ135を読み出し、RAM105に転送させる。そして、CPU103は、RAM105に転送されたゲームプログラム133を実行することにより、以下に記載する処理を実現する。

【0090】

なお、家庭用ゲーム装置101で行われる制御及び処理の中には、CPU10

3以外の回路がCPU103と協働して実際の制御及び処理を行っている場合もある。説明の便宜上、以下では、CPU103が関係する制御及び処理は、CPU103が直接的に実行しているものとして説明する。

【0091】

また、画像処理やゲームを実行するために必要なゲームプログラム133やデータ135は、実際にはCPU103からの命令に従って処理の進行状況に応じて順次CD-ROM131から読み出されてRAM105に転送される。しかし、以下に示す説明では、発明の理解を容易にするため、CD-ROM131からのデータの読み出しや、RAM105への転送に関する記述は省略している。

【0092】

もし画像処理やゲームを実行するために必要なゲームプログラム133やデータ135がHDD107に格納されている場合には、CPU103からの命令に従って処理の進行状況に応じて順次HDD107から読み出されてRAM105に転送される。しかし、以下に示す説明では、上で述べたのと同様の理由で、HDD107からのデータの読み出しや、RAM105への転送に関する記述は省略している。

【0093】

図12に本実施の形態に係るメインフローを示す。最初に初期設定が行われる(ステップS1)。この初期設定では、図2乃至図5、図8及び図9において示したようなデータをRAM105に読み出す。また、フレームカウンタを0にリセットしておく。次に、仮想空間内の状態設定を行う(ステップS3)。仮想空間内の状態設定では、例えばユーザによるキーパッド161の操作により三次元オブジェクトのアニメーションの開始が命令されたり、同じくユーザによるキーパッド161の操作により視点位置が変更される場合に、それに応じて仮想空間内の状態を変更する処理である。

【0094】

そして本発明に係るオブジェクト変形処理が実施される(ステップS5)。これについては後に詳しく述べる。変形された三次元オブジェクトに対して描画処理が行われる(ステップS7)。光源計算、透視変換等を実施して、フレームバ

ッファ112に描画する。本実施の形態では描画処理は従来と変わらない。次にフレーム・カウンタ1100が1インクリメントされる(ステップS9)。そして処理が終了か否かが判断される(ステップS11)。フレーム・カウンタ1100の値が処理の終了を示している場合には処理を終了する。フレーム・カウンタ1100の値が処理の終了を示していない場合にはステップS3に戻る。

【0095】

オブジェクト変形処理の詳細を図13に示す。図12の説明で分かるように、このオブジェクト変形処理を実施する前に、フレーム番号は分かっている。まず最初に、三次元オブジェクトの元形状を取得する(ステップS21)。ここではRAM105にコピーされた頂点座標リスト200をコピーしてRAM105内に頂点バッファ850を生成する。そして、全ジョイントについて変形処理を実施したか判断する(ステップS23)。全ジョイントについて変形処理が実施されていれば処理を終了する。全てのジョイントについて変形処理が終了しているわけではない場合にはステップS25に移行する。

【0096】

ステップS25では、アニメーション・データ700からジョイント・インデックスを取得する(ステップS25)。そして取得したジョイント・インデックスに係るカレントジョイント情報を取得する(ステップS27)。カレントジョイント情報とは、当該ジョイントの回転角(R_x , R_y , R_z)である。アニメーション・データ700の説明でも述べたが、より上位の階層のジョイントから順番に取得されるようにする。すなわち、より上位の階層のジョイントについては処理が終了しているようにする。このためアニメーション・データ700においてより上位の階層のジョイントは上のほうに並べられる。

【0097】

次に取得されたジョイント・インデックスに対応付けられた頂点グループ全てについて移動処理を終了したか判断する(ステップS29)。もし、取得されたジョイント・インデックスに対応付けられた頂点グループ全てについて移動処理を終了している場合には、次のジョイント・インデックスの処理に移行するためステップS23に戻る。一方、取得されたジョイント・インデックスに対応付け

られた全ての頂点グループについて移動処理が終了しているわけではない場合には、未処理の頂点グループ番号をジョイント・データ 6 0 0 から取得する（ステップ S 3 1）。

【 0 0 9 8 】

次に、頂点グループ・リスト 5 0 0 を参照して、取得した頂点グループ番号に属する頂点について全て移動処理が終了したか判断する（ステップ S 3 3）。頂点グループ・リスト 5 0 0 は頂点グループ番号の情報を含んでいない。但し、頂点グループ・リスト 5 0 0 を上から走査し、頂点数（メンバ数）の行 5 2 0 乃至 5 2 4 の出現回数により頂点グループ番号を把握することができる。すなわち、頂点数の行 5 2 0 乃至 5 2 4 の出現回数 - 1 が頂点グループ番号を意味するため、頂点数の行 5 2 0 乃至 5 2 4 の出現回数 - 1 = 取得した頂点グループ番号となるまで頂点グループ・リスト 5 0 0 を上から走査すれば、取得した頂点グループ番号に関する頂点の情報を得ることができる。

【 0 0 9 9 】

もし、取得した頂点グループ番号に属する頂点について全て移動処理が終了している場合には、ステップ S 2 9 に戻る。一方、取得した頂点グループ番号に属する全ての頂点について移動処理が終了しているわけではない場合には、頂点グループ・リスト 5 0 0 から未処理の頂点インデックスを取得し、同様に頂点グループ・リスト 5 0 0 から当該頂点の重み w を取得する（ステップ S 3 5）。

【 0 1 0 0 】

そしてカレントジョイント情報（現フレームにおける回転角）と重み w に基づいて頂点バッファの座標情報を変更する（ステップ S 3 7）。ステップ S 3 7 では、上で述べたクォータニオンの球状線形補間の処理が行われる。すなわち、現フレームにおけるジョイントの回転角によるクォータニオン q_1 と単位クォータニオン q_i とを重み w で球状線形補間する。球状線形補間により求められたクォータニオン q から本ジョイントにおける変換マトリックス R が上の（5）の式で求められる。

【 0 1 0 1 】

元の頂点座標 v からジョイントの回転による回転後の頂点座標 v_a を求めるた

めの全体のマトリックスは、変換マトリックスRの他に、親ジョイントからの相対座標を表すマトリックスT、基準回転角を表すマトリックスJ、親ジョイントの変換マトリックスBを用いて、 $M = R J T B$ となる。すなわち、最終的に求めるべき頂点座標 v_a は以下の式で求められる。

$$v_a = v M$$

【0102】

計算された $v_a = (x_a, y_a, z_a)$ を頂点バッファ850に書き込む。また、当該頂点の法線ベクトルも法線ベクトル・リスト300から取り出して、変換マトリックスMの回転成分のみを用いて変換した法線ベクトルを、法線ベクトル・バッファ870に書き込む。これにより、頂点の座標及び法線ベクトルの値が更新される。更新された頂点座標及び法線ベクトルの値はそれぞれ頂点バッファ850及び法線ベクトル・バッファ870に格納され、頂点バッファ850及び法線ベクトル・バッファ870の内容に従って三次元オブジェクトがフレームバッファ112に描画される。

【0103】

ステップS37の後にはステップS33に戻り、頂点グループの全ての頂点について移動処理が終了するまでステップS35及びS37が実施される。

【0104】

なお、ステップS37の処理では、マトリックスMを用いて法線ベクトルを変換しているが、これはリアルタイムで処理を行うための近似手段である。ポリゴンのエッジ情報（隣接するポリゴンと成す角）等を考慮した上で法線ベクトルを生成すれば、より正確な法線ベクトルを求めることができる。

【0105】

上で述べたステップS37の処理では全ての頂点について変換マトリックスMを計算していた。しかし、全てについて変換マトリックスMを計算する必要は無い。同じジョイントに対応する頂点の場合には、頂点に予め定義されている重みが同じであれば、前に計算された変換マトリックスMはそのまま使用できる。以下では変換マトリックスBのバッファ及び変換マトリックスMのキャッシュについて説明する。

【 0 1 0 6 】

図 1 4 に親ジョイントの変換マトリックス B のマトリックス・バッファ 8 0 0 を示す。アニメーション・データ 7 0 0 の各タイム・フレームでは、ルート・ジョイントから最下位の階層のジョイントへ順番に処理される。ルート・ジョイントを除く全てのジョイントは親ジョイントからの影響を受けるため、親ジョイントの重み 1 0 0 % の変換マトリックス B が必要となる。また親ジョイントに対応する頂点の処理時においても、当該親ジョイントに対応する頂点の重みが 1 0 0 % であれば、そのまま変換マトリックス B は使用可能である。

【 0 1 0 7 】

よって、マトリックス・バッファ 8 0 0 では、ジョイント毎に重み 1 0 0 % における変換マトリックスのデータを保持しておく。ジョイント・インデックスの欄 8 1 0 には、ルート・ジョイント以下のジョイント・インデックスが格納される。マトリックス・データの欄 8 2 0 には、各ジョイントの変換マトリックスのデータが格納される。上では $M = R J T B$ であった。基本的には同じ式で計算できる。ここでは重み 1 0 0 % でクォータニオンを使用しないで求めた変換マトリックス R_a と、親ジョイントの更に親ジョイントの重み 1 0 0 % の変換マトリックスを B_a とする。求めるべき変換マトリックス B は $B = R_a J T B_a$ となる。

【 0 1 0 8 】

変換マトリックス M のキャッシュについてここでは 2 種類の実施の形態を示しておく。図 1 5 には直前に計算した頂点の重みの値を記憶する欄 9 1 0 と当該重みに対応する変換マトリックス M の欄 9 2 0 を設けたマトリックス・キャッシュ 9 0 0 を示す。同じジョイントについて重みが同じ頂点が複数あれば、1 度計算された変換マトリックス M は再利用可能である。

【 0 1 0 9 】

よって、新たな頂点を移動処理する場合（図 1 3 : ステップ S 3 7）には、まずマトリックス・キャッシュ 9 0 0 を参照する。新たな頂点の重みの値と、マトリックス・キャッシュ 9 0 0 の重みの欄 9 1 0 に格納された重みの値が一致するか判断する。もし、新たな頂点の重みの値と、マトリックス・キャッシュの重みの欄 9 1 0 に格納された重みの値が一致する場合には、変換マトリックス M の欄

920に格納された変換マトリックスMのデータは、新たな頂点に対して使用可能である。よって、マトリックス・キャッシュ900から読み出して、読み出した変換マトリックスMにより当該頂点を移動させる。一方、新たな頂点の重みの値と、マトリックス・キャッシュ900の重みの欄910に格納された重みの値が一致しない場合には、上で述べたのと同じように新たに変換マトリックスMを計算する。そして新たな頂点の重みと、計算された変換マトリックスMとをマトリックス・キャッシュ900に格納する。

【0110】

ジョイント毎に又は頂点グループ毎に、頂点インデックスが頂点の重みのデータによりソートされていれば本手法は有効である。なお、図5の頂点グループ・リスト500では頂点インデックスが頂点の重みのデータによりソートされている様子が示されている。

【0111】

図15では1つの変換マトリックスMのデータしかキャッシュできなかったが、図16では複数の変換マトリックスMのデータをキャッシュする方法が示されている。マトリックス・キャッシュ1000では、1つの重み毎に、重みの値の欄1010、1020及び1030、変換マトリックスのデータの欄1012、1022及び1032、次のデータへのポインタ1014、1024及び1034が含まれる。

【0112】

図16の例では、重み0.4の場合と、重み0.2の場合と、重み0.8の場合の変換マトリックスが格納されている。なお、最後のデータについては次のデータへのポインタではなくnullが格納されている。

【0113】

図16のようなマトリックス・キャッシュ1000が用意されている場合には、新たな頂点を移動処理する場合（図13：ステップS37）には、まずマトリックス・キャッシュ1000を参照する。新たな頂点の重みの値と、マトリックス・キャッシュ1000の重みの欄1010乃至1030のいずれかに格納された重みの値が一致するか判断する。

【0114】

もし、新たな頂点の重みの値と、マトリックス・キャッシュの重みの欄1010乃至1030のいずれかに格納された重みの値が一致する場合には、変換マトリックスMの欄1012乃至1032に格納された、対応する変換マトリックスMのデータは、新たな頂点に対して使用可能である。よって、マトリックス・キャッシュ1000から読み出して、読み出した変換マトリックスにより当該頂点を移動させる。

【0115】

一方、新たな頂点の重みの値と、マトリックス・キャッシュ1000の重みの欄1010乃至1030に格納された重みの値が全く一致しない場合には、上で述べたのと同じように新たに変換マトリックスMを計算する。そして新たな頂点の重みと、計算された変換マトリックスMとをマトリックス・キャッシュ1000に新たなデータとして付加する。

【0116】

図16のようなマトリックス・キャッシュ1000を使用する場合には、各頂点の重みが段階的に定義されていると、マトリックス・キャッシュ1000のヒット率が高まるので、速度・容量共に効率が上がる。段階的な重みとは、例えば重みが0.0から1.0の範囲で設定可能である場合、0.2単位でのみ重みを定義することができるようにするものである。例えば0.1や0.79等の0.2の単位からはずれる重みを設定不可能とする。

【0117】

図16のようなマトリックス・キャッシュ1000を使用する場合には、図15のようなマトリックス・キャッシュ900のように頂点インデックスが頂点の重みによりソートされていなくとも有効に作用する。但し、マトリックス・キャッシュ1000には複数のデータが存在するのでサーチの処理が必要となる。

【0118】

マトリックス・キャッシュ1000のサーチ処理を高速化するために、以下のような処理を行うことができる。図17では、高速にサーチ可能な複数の変換マトリックス・キャッシュの構造を模式的に示している。図17に示したマトリッ

クス・キャッシュ2000では、1つの重み毎のデータに、重みの欄2010、2020及び2030、変換マトリックスのデータ欄2012、2022及び2032、次のデータへのポインタ(next)2014、2024及び2034、前のデータへのポインタ(prev)2016、2026及び2036が含まれている。

【0119】

図17の例では、重み0.2の場合と、重み0.4の場合と、重み0.8の場合の変換マトリックスのデータが格納されている。なお、重み0.2の場合の変換マトリックスは先頭のデータであり、前のデータへのポインタ2016にはnullが格納されている。また、重み0.8の場合の変換マトリックスは最後尾のデータであり、次のデータへのポインタ2034にはnullが格納されている。

【0120】

マトリックス・キャッシュ2000は、常に重み値によってソートされている。すなわち、ある重みの変換マトリックスのデータを追加する場合には、追加するデータの重みより低い重みのデータの次、追加するデータの重みより高い重みのデータの前にデータが追加される。

【0121】

次のデータへのポインタ2014には、重み0.4のデータの位置を示すポインタが格納される。次のデータへのポインタ2024には、重み0.8のデータの位置を示すポインタが格納される。前のデータへのポインタ2036には、重み0.4のデータの位置を示すポインタが格納される。前のデータへのポインタ2026には、重み0.2のデータの位置を示すポインタが格納される。

【0122】

例えば、図17に示したデータが格納されている状態で、更に重み0.6の場合の変換マトリックスのデータを追加するには、重み0.4のデータの次、重み0.8のデータの前に重み0.6のデータが格納される。その際、重み0.4の次のデータへのポインタ2024には、重み0.6のデータの位置を示すポインタが格納される。重み0.8の前のデータへのポインタ2024には、重み0.

6 のデータの位置を示すポインタが格納される。

【 0 1 2 3 】

また、図 1 7 の例ではマトリックス・キャッシュ 2 0 0 0 のほかに、直前に参照したデータを指し示す参照データポインタ 2 1 0 0 が設けられている。参照データポインタ 2 1 0 0 には、マトリックス・キャッシュ 2 0 0 0 にデータを追加した場合、及びマトリックス・キャッシュ 2 0 0 0 のデータを参照した場合に、そのデータへのポインタが格納される。

【 0 1 2 4 】

図 1 7 に示したデータ構造とし、図 5 の頂点インデックスを頂点の重みのデータによりソートしておく。新たな頂点を移動処理する場合（図 1 3 : ステップ S 3 7）には、まず参照データポインタ 2 1 0 0 が参照される。そして、マトリックス・キャッシュ 2 0 0 0 内の参照データポインタ 2 1 0 0 で指し示されたデータが参照され、処理すべき頂点の重みと参照したデータの重みが比較される。

【 0 1 2 5 】

比較の結果、2 つの重みが一致する場合には、参照したデータの変換マトリックスにより、頂点の座標が変換される。比較の結果、処理すべき頂点の重みが参照したデータの重みより大きい場合には、参照したデータの次のデータが参照される。比較の結果、処理すべき頂点の重みが参照したデータの重みより小さい場合には、参照したデータの前のデータが参照される。

【 0 1 2 6 】

その後、参照したデータの重みと処理すべき頂点の重みとの比較が順次行われる。処理すべき頂点の重みが参照したデータの重みより大きいとして次のデータを参照し、次のデータと比較の結果、処理すべき頂点の重みが参照したデータの重みより小さいと判断された場合には、マトリックス・キャッシュ 2 0 0 0 内に目的の変換マトリックスが無いと判断される。この場合、最後に参照したデータの前に、新たに計算された重みの変換マトリックスのデータが格納される。

【 0 1 2 7 】

また、処理すべき頂点の重みが参照したデータの重みより大きいと判断されたが、次のデータへのポインタが `null` である場合、マトリックス・キャッシュ

2 0 0 0 内に目的の変換マトリックスが無いと判断される。この場合、最後に参照したデータの次に、新たに計算された重みの変換マトリックスのデータが格納される。

【 0 1 2 8 】

同様に、処理すべき頂点の重みが参照したデータの重みより小さいとして、前のデータを参照し、前のデータとの比較の結果、処理すべき頂点の重みが参照したデータの重みより大きいと判断された場合には、マトリックス・キャッシュ 2 0 0 0 内に目的の変換マトリックスは無いと判断される。この場合、最後に参照したデータの次に、新たに計算された重みの変換マトリックスのデータが格納される。

【 0 1 2 9 】

また、処理すべき頂点の重みが参照したデータの重みより小さいと判断されたが、前のデータへのポインタが `null` である場合、マトリックス・キャッシュ 2 0 0 0 内に目的の変換マトリックスのデータが無いと判断される。この場合、最後に参照したデータの前に、新たに計算された重みの変換マトリックスのデータが格納される。

【 0 1 3 0 】

図 1 7 に示したように、参照データポインタ 2 1 0 0 を設け、図 5 の頂点インデックスを頂点の重みのデータによりソートしておくことで、目的のデータを早い段階で検出することができ、処理の高速化が図れる。

【 0 1 3 1 】

また、マトリックス・キャッシュ 2 0 0 0 内のデータ配列を重みによってソートし、前のデータと次のデータとの双方に対してデータのポインタを設けたことで、全てのデータをサーチする必要がなくなる。すなわち、データがソートされているため、最初に参照したデータの重みの比較結果から、重みの昇順若しくは降順の何れかの順番でサーチを行えばよい。その結果、サーチ処理が高速化される。

【 0 1 3 2 】

以上のようにマトリックス・キャッシュを導入することにより、変換マトリッ

クスMの計算処理の量が削減され、処理速度が高速化される。

【0133】

以上述べたように、複数の頂点により形状が定められる三次元オブジェクトを変形する場合には、まず三次元オブジェクトに対応しており、複数のジョイントにより構成され、且つ複数の頂点の各々が複数のジョイントのいずれか1つに対応付けられている仮想スケルトンにおける各ジョイントの回転角をタイム・フレーム毎に取得する（ステップS27）。そして、各頂点に対応し且つ取得されたジョイントの回転角を、当該頂点に予め定義された重みに従って変更し、変更された回転角に従って頂点をタイム・フレーム毎に移動する（ステップS37）。

【0134】

頂点に予め定義された重みに従って当該頂点に対応するジョイントの回転角を変更することにより、当該頂点の回転角が求められる。頂点の回転角に従って頂点を移動する。

【0135】

よって、仮想三次元空間において1つの頂点について2つの位置情報を計算することなく、三次元オブジェクトを滑らかに変形することができる。特にクォータニオンを使用することにより、最短距離にて頂点の回転がなされる。

【0136】

また近年、より写実的なCG画像を得るために、三次元オブジェクトに定義される頂点数は非常に増加している。本発明によれば1つの頂点につき一回の頂点変換で済むため、頂点数が増加しても処理量の増加量は少なくて済む。

【0137】

すなわち、変形後の三次元オブジェクトの滑らかさを要求しない場合であっても、形状を変形する際には各頂点に1回ずつの頂点変換は必要である。従って、本発明を適用して滑らかな形状が保たれるようにしても、頂点変換の回数は変わらない。

【0138】

なお、本発明を実施する場合、ジョイントの回転角と頂点に与えられた重みに応じて、変換マトリックスを算出しなければならない。但し、変換マトリック

スの算出処理は、図15乃至図17に示したマトリックス・キャッシュを用いることで、頂点毎に行う必要が無くなる。従って、本発明では、頂点数が増加しても、変形後の形状を滑らかにするために必要な処理の増加量を抑制することが可能である。

【0139】

また、上記実施の形態では、ジョイントの回転角と同じ回転軸で各頂点を回転移動させている。そのため、ジョイントから頂点までの距離を保ったまま、頂点を移動させることができる。例えば、人間の膝には骨があるため、人間を模した三次元オブジェクトの膝の部分のジョイントから膝頭の頂点までの距離は、一定の距離が保たれている必要がある。本発明を適用することで、膝の部分のジョイントの変形を行っても、膝の部分の膨らみを保ったままの変形が可能となる。

【0140】

(その他の実施の形態)

(1) アニメーション・データ700では、全てのフレームについて回転角のデータが格納されていることを前提としていた。しかし、必ずしも全てのフレームについて回転角のデータが格納されている必要は無い。データの無いフレームについては、前後のフレームの回転角を補間することにより求めることができる。

【0141】

(2) 上の実施の形態では三角形ポリゴンを前提に説明したが、本発明は、四角形以上の多角形ポリゴンであっても、頂点を移動することにより三次元オブジェクトを変形させるモデルにおいては適用することができる。

【0142】

(3) 上の実施の形態では変換マトリックスを生成する例を示しているが、マトリックスを使用しないで処理することも可能である。但し、変換マトリックスは平行移動も同時に取り扱うことができるので有利である。

【0143】

(4) 上の実施の形態では、重みの反映をクォータニオン及び球状線形補間を用いて行ったが、球状線形補間に変えて単なる線形補間を用いることも可能である。ここでいう単なる線形補間とは、2つのクォータニオン間の変数同士で、重み

に応じた線形補間を行う処理である。単なる線形補間は球状線形補間に比べて簡単な処理で実施可能であるため、単なる線形補間を用いることにより処理が簡略化される。なお、単なる線形補間を用いたとしても、角度の変化がある程度以下に抑えられれば、三次元オブジェクトがビデオゲームの画面に不自然な形状で表示されることは無い。

【 0 1 4 4 】

(5) 使用するハードウェアの変更

図 1 は一例であって、様々な変更が可能である。通信インターフェース 1 1 5 を備えるか否かは任意である。本発明は直接サウンド処理には関係しないので、サウンド処理部 1 0 9 を備えている必要は無い。

【 0 1 4 5 】

また、CD-ROM は記録媒体の一例であって、ROM のような内部メモリ、CD-ROM、DVD-ROM、メモ리카ートリッジ、フロッピーディスク、磁気ディスク、DVD-RAM 等の他の記録媒体であってもよい。その場合には CD-ROM ドライブ 1 1 3 を、対応する媒体で読み出し可能なドライブにする必要がある。

【 0 1 4 6 】

さらに、以上は本発明をコンピュータ・プログラムにより実装した場合であるが、コンピュータ・プログラムと電子回路などの専用の装置の組み合わせ、又は電子回路などの専用の装置のみによっても実装することは可能である。

【 0 1 4 7 】

以上、本発明を実施の形態に基づいて具体的に説明したが、本発明は上記実施の形態に限定されるものではない。要旨を逸脱しない範囲で適宜変更可能である。例えば、上記実施の形態では、家庭用ゲーム機をプラットフォームとして本発明を実現した場合について述べたが、本発明は通常のコンピュータ、アーケードゲーム機などをプラットフォームとして実現しても良い。また、携帯情報端末、カーナビゲーション・システム等をプラットフォームにして実現することも考えられる。

【 0 1 4 8 】

また、本発明を実現するためのプログラムやデータは、コンピュータやゲーム機に対して着脱可能なCD-ROM等の記録媒体により提供される形態に限定されない。すなわち、本発明を実現するためのプログラムやデータを、図1に示す通信インターフェース115、通信回線141を介して接続されたネットワーク151上の他の機器側のメモリに記録し、プログラムやデータを通信回線141を介して必要に応じて順次RAM105に格納して使用する形態であってもよい。

【0149】

(表示例)

図18に、基準姿勢における人型の三次元オブジェクトの脚の部分のポリゴン・モデルを示す。参考のため、仮想スケルトンのジョイントJ1及びボーンB1乃至B3（符号は右の足のみ）を示している。図18では、ジョイントJ1を回転させると影響が大きく表われる部分A及びBを示しておく。部分Aの3つの四角形ポリゴンa1乃至a3、部分Bの5つの四角形ポリゴンb1乃至b5の変化を見る。

【0150】

図19に、本発明を使用してジョイントJ1を回転させた場合を示す。上で述べたように、三次元オブジェクトの各頂点は、対応するジョイントの回転角を頂点に予め定義された重みによって変更することにより得られる頂点用の回転角により移動される。図18の部分A及び部分Bのポリゴンの各頂点はジョイントJ1に対応しており、部分Aは部分Aaに、部分Bは部分Baに移動した。

【0151】

部分Aaのポリゴンa1zは図18のポリゴンa1に、ポリゴンa2zは図18のポリゴンa2に、ポリゴンa3zは図18のポリゴンa3にそれぞれ対応する。ジョイントJ1の回転に大きく影響を受けたポリゴンa2zは、その面積が非常に小さくなっている。図19では、ポリゴンa1z及びa3zもそれぞれ対応するポリゴンa1及びa3より小さくなっている。

【0152】

部分Baのポリゴンb1zは図18のポリゴンb1に、ポリゴンb2zは図18のポ

リゴン b2 に、ポリゴン b3z は図 18 のポリゴン b3 に、ポリゴン b4z は図 18 のポリゴン b4 に、ポリゴン b5z は図 18 のポリゴン b5 にそれぞれ対応する。部分 Aa とは対照的に部分 Bz は引き伸ばされる部分である。ジョイント J1 の回転に大きく影響を受けたポリゴン b3z 及び b4z は、対応する b3 及び b4 より大きくなっている。また、ポリゴン b1z、b2z 及び b5z も、対応するポリゴン b1、b2 及び b5 より大きくなっている。

【0153】

図 18 のポリゴン・モデルを描画すると図 20 のような画像を得ることができる。滑らかな三次元オブジェクトが表現できている。一方、図 19 のポリゴン・モデルを描画すると図 21 のような画像を得ることができる。本発明を使用して三次元オブジェクトを変形しても、図 21 で分かるように、部分 Aa でポリゴンの頂点が三次元オブジェクトにめり込むことはなく、ポリゴンが大きくなりすぎたりしない。全体として三次元オブジェクトは滑らかに描画されるようになる。

【0154】

【発明の効果】

以上述べたように本発明により、ジョイントの回転角と頂点に定義された重みとに基づいて頂点の回転角度を算出し、算出された角度だけその頂点を回転させることで、変形後の三次元オブジェクトの形状を滑らかにすることができる。しかも、各頂点に対する頂点変換は 1 回しか行われず、三次元オブジェクトの頂点数が増加した場合の処理の増加量を、少なく抑えることが可能となる。

【図面の簡単な説明】

【図 1】

家庭用ゲーム機のブロック構成図である。

【図 2】

頂点座標リストの一例を示す図である。

【図 3】

法線ベクトル・リストの一例を示す図である。

【図 4】

三角形ポリゴン・リストの一例を示す図である。

【図 5】

頂点グループ・リストの一例を示す図である。

【図 6】

三次元オブジェクトと仮想スケルトンの一例を示すための模式図である。

【図 7】

図 6 における仮想スケルトンのジョイントの階層構造を示すための模式図である。

【図 8】

ジョイント・データの一例を示す図である。

【図 9】

アニメーション・データの一例を示す図である。

【図 10】

本発明の動作原理を説明するための模式図である。(a) はジョイントを回転させる前の状態を示す。(b) は全ての頂点の重みを 100% としてジョイントを回転させた場合の状態を示す。(c) は特開平 10-74270 号公報記載の技術を用いてジョイントを回転させた場合の状態を示す。(d) は本発明を用いてジョイントを回転させた場合の状態を示す。

【図 11】

球状線形補間を説明するための模式図である。

【図 12】

本発明の処理のメインフローを示すフローチャートである。

【図 13】

本発明の三次元オブジェクト変形処理のフローチャートである。

【図 14】

マトリックス・バッファの一例を示す図である。

【図 15】

マトリックス・キャッシュの一例を示す図である。

【図 16】

マトリックス・キャッシュの一例を示す図である。

【図17】

マトリックス・キャッシュの一例を示す図である。

【図18】

基本姿勢における三次元オブジェクトのポリゴン・モデルを示す図である。

【図19】

足を曲げた状態における三次元オブジェクトのポリゴン・モデルを示す図である。

【図20】

図18のポリゴン・モデルを描画した場合の画像例を示す図である。

【図21】

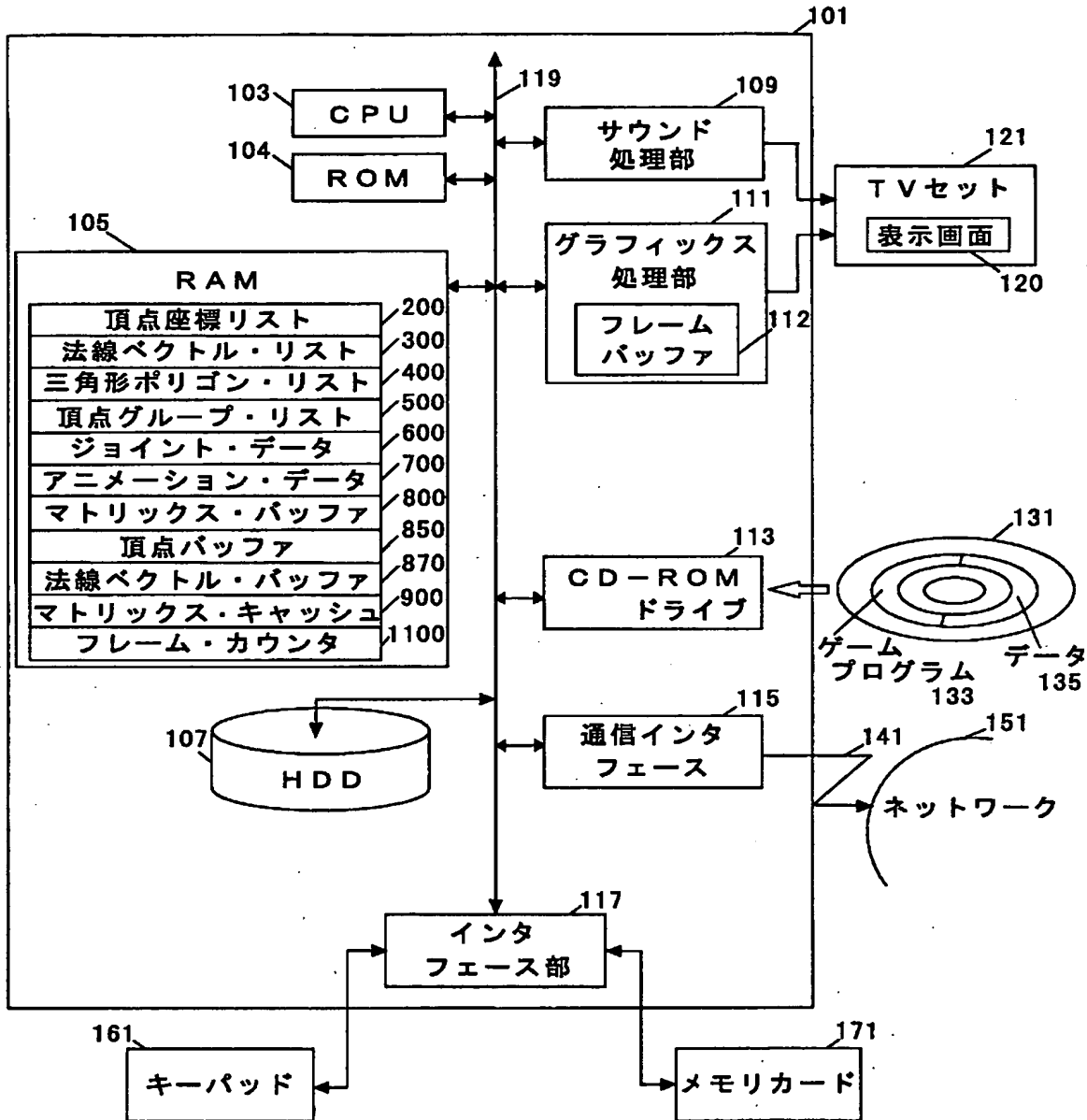
図19のポリゴン・モデルを描画した場合の画像例を示す図である。

【符号の説明】

101	家庭用ゲーム機	103	CPU	104	ROM
105	RAM	107	HDD	109	サウンド処理部
112	フレームバッファ	111	グラフィックス処理部		
113	CD-ROMドライブ	115	通信インターフェース		
117	インターフェース部	119	内部バス	121	TVセット
120	表示画面	131	CD-ROM	141	通信媒体
151	ネットワーク	161	キーパッド	171	メモリカード
200	頂点座標リスト	300	法線ベクトル・リスト		
400	三角形ポリゴン・リスト	500	頂点グループ・リスト		
600	ジョイント・データ	700	アニメーション・データ		
800	マトリックス・バッファ	850	頂点バッファ		
870	法線ベクトル・バッファ	1100	フレーム・カウンタ		
900, 1000, 2000	マトリックス・キャッシュ				

【書類名】 図面

【図 1】



【図 2】

200

210 220

index	座標 (x, y, z)
vtx0	(0, 0, 0)
vtx1	(0, 0, 100)
vtx2	(0, 100, 0)
vtx3	(0, 100, 100)
vtx4	(100, 0, 0)
⋮	⋮

【図 3】

300

310 320

index	法線ベクトル (x, y, z)
nm10	(1, 0, 0)
nm11	(-1, 0, 0)
nm12	(0, 0, -1)
nm13	(-1, -1, 0)
nm14	(0, -1, 0)
⋮	⋮

【図 4】

index	頂点0	頂点1	頂点2
plg0	vtx0/nml0	vtx1/nml2	vtx2/nml3
plg1	vtx0/nml1	vtx2/nml3	vtx3/nml4
⋮	⋮	⋮	⋮

【図 5】

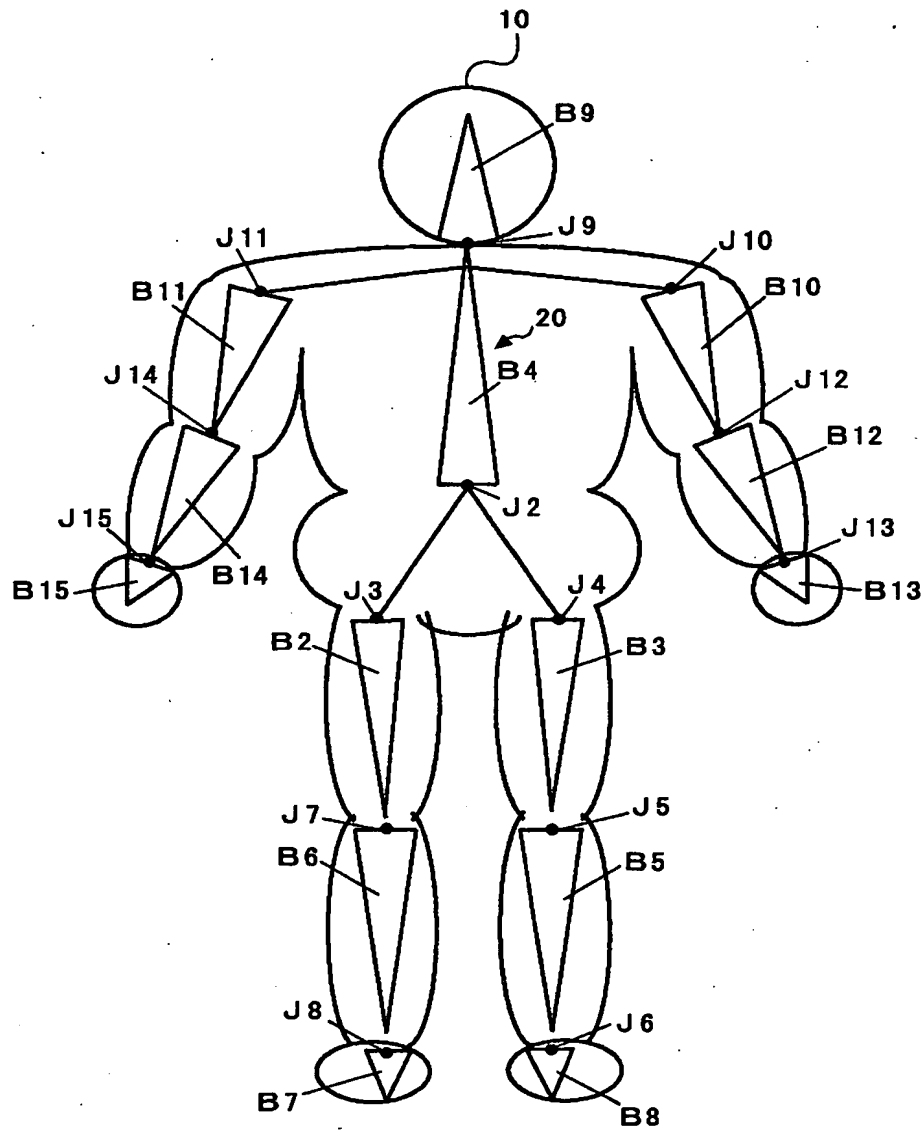
頂点グループリスト									
510	グループ数		3						
520	メンバ数		5						
530	頂点	vtx0	重み (540)	1.0	法線数 (570)	2	法線 (586)	nm10	nm11
532	頂点	vtx5	重み (542)	1.0	法線数 (572)	1	法線 (588)	nm110	
534	頂点	vtx1	重み (544)	0.7	法線数 (574)	1	法線 (590)	nm12	
536	頂点	vtx2	重み (546)	0.6	法線数 (576)	2	法線 (592)	nm13	nm11
538	頂点	vtx3	重み (548)	0.5	法線数 (578)	1	法線 (594)	nm14	
522	メンバ数		2						
550	頂点	vtx4	重み (560)	0.3	法線数 (580)	1	法線 (596)	nm15	
552	頂点	vtx6	重み (562)	0.2	法線数 (582)	3	法線 (598)	nm16	nm18 nm19
524	メンバ数		1						
554	頂点	vtx7	重み (564)	0.5	法線数 (584)	1	法線 (599)	nm17	

頂点グループ0

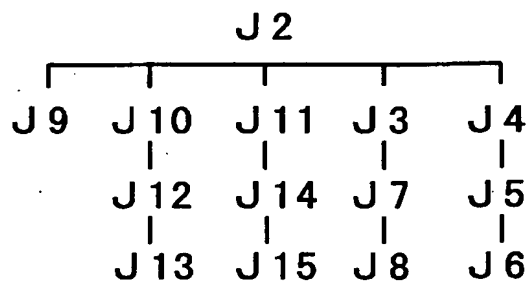
頂点グループ1

頂点グループ2

【図6】



【図7】



【図 8】

600

610	ルートindex	root
612	基準平行移動量 (x, y, z)	(0, 0, 0)
614	基準回転角 (Rx, Ry, Rz)	(0, 0, 90)
616	ジョイント数	4
620	ジョイントindex	joint0
622	親index	root
624	基準平行移動量 (x, y, z)	(90, 20, 0)
626	基準回転角 (Rx, Ry, Rz)	(0, 0, 135)
628	頂点グループ数	2
630	頂点グループ番号	0
632	頂点グループ番号	2
640	ジョイントindex	joint1
642	親index	joint0
644	基準平行移動量 (x, y, z)	(40, 0, 0)
646	基準回転角 (Rx, Ry, Rz)	(0, 0, 45)
648	頂点グループ数	1
650	頂点グループ番号	1

joint0

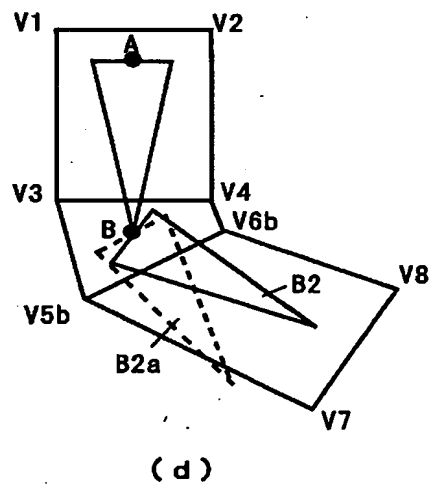
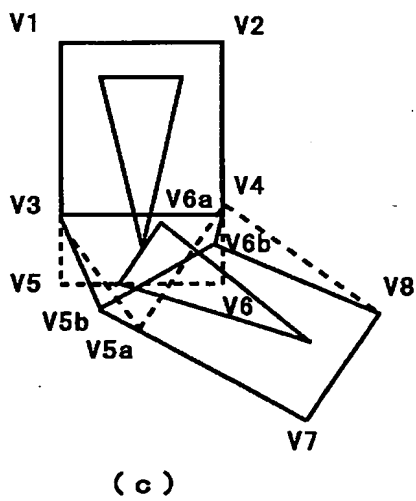
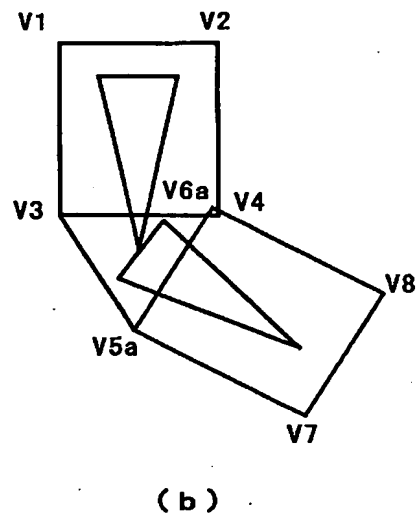
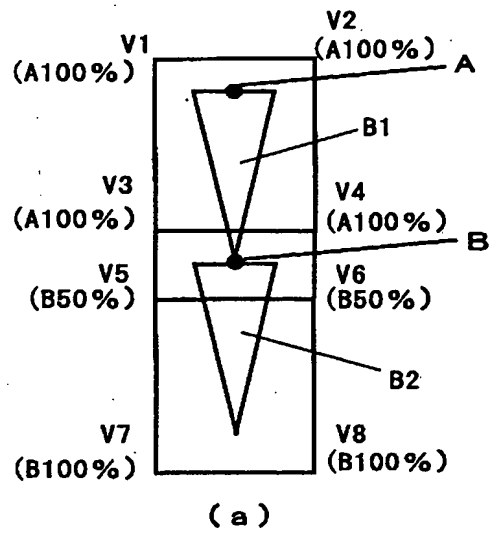
joint1

【図 9】

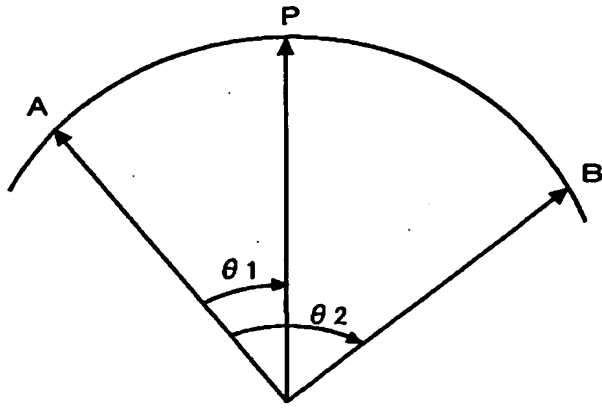
710 ジョイント数	720 フレーム数	700A 700
2	10	

730 ジョイント index	740 回転角 (Rx, Ry, Rz)	700B
Joint0	(0, 0, 0)	
Joint1	(0, 0, 0)	フレーム0
Joint0	(10, 0, 135)	
Joint1	(-10, 5, 45)	フレーム1
⋮	⋮	
Joint0	(90, 0, 150)	フレーム9
Joint1	(-90, 45, 45)	

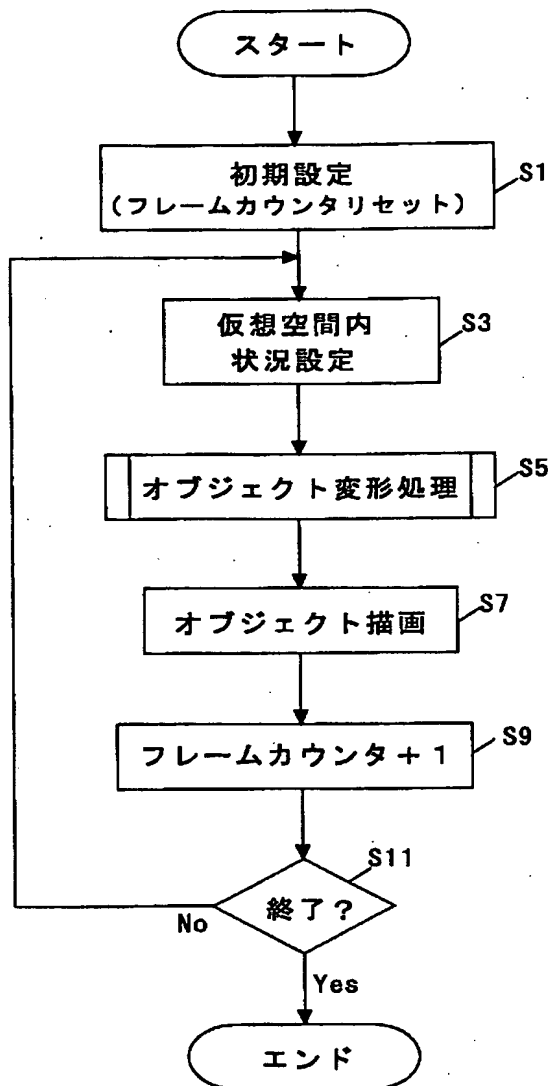
【図 10】



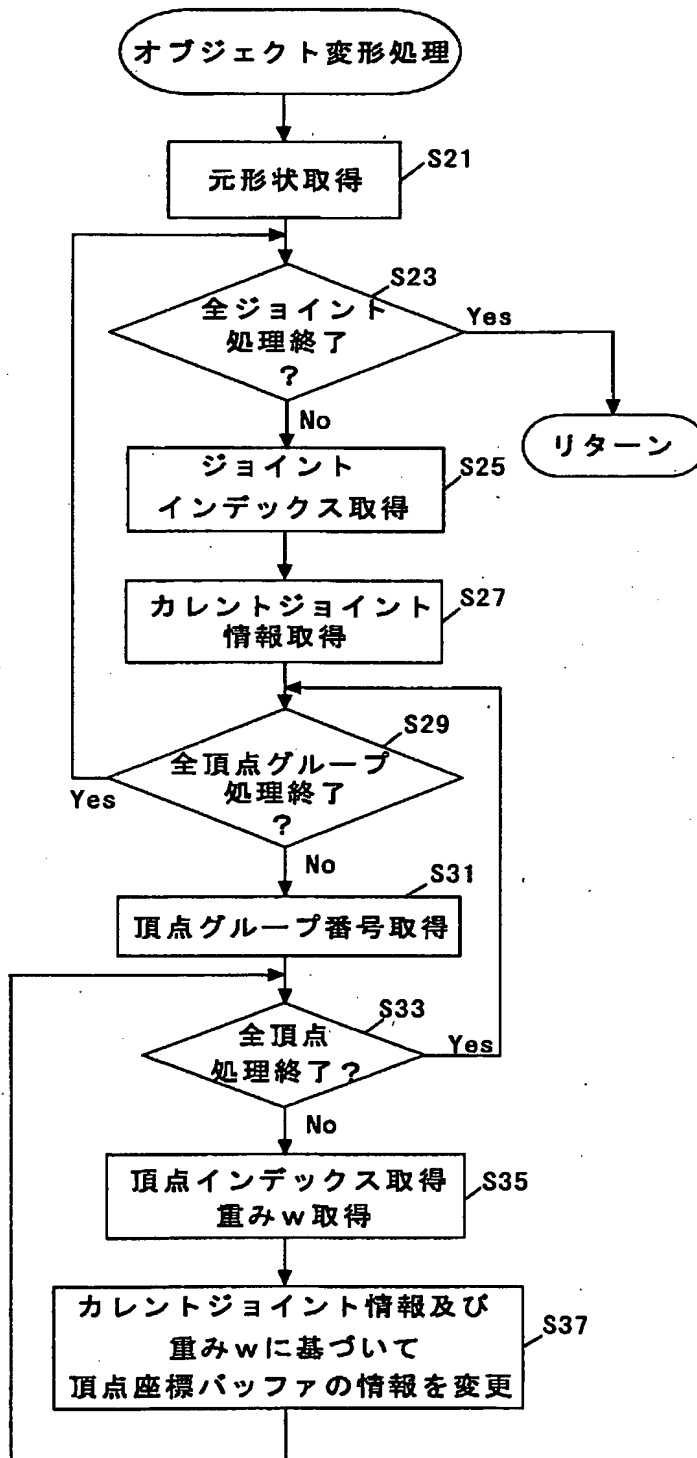
【図 11】



【図 12】



【図13】



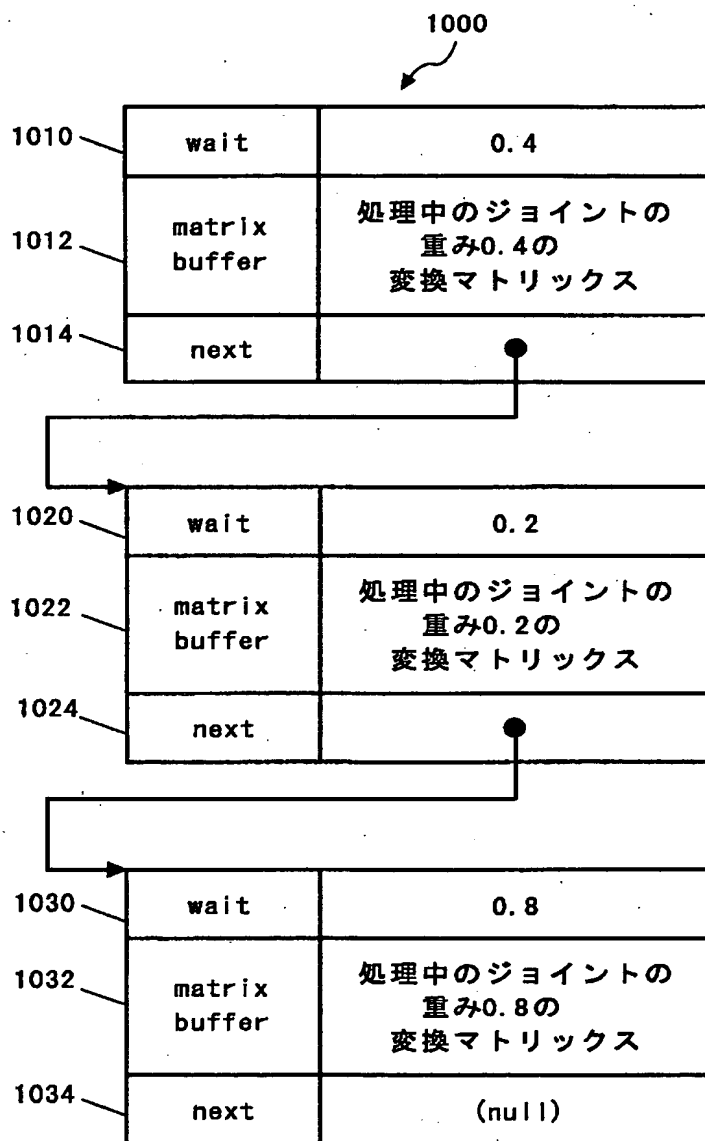
【図 1 4】

ジョイント index	マトリックス データ
root	ルート 変換マトリックス
joint0	joint0の重み100%の 変換マトリックス
joint1	joint1の重み100%の 変換マトリックス

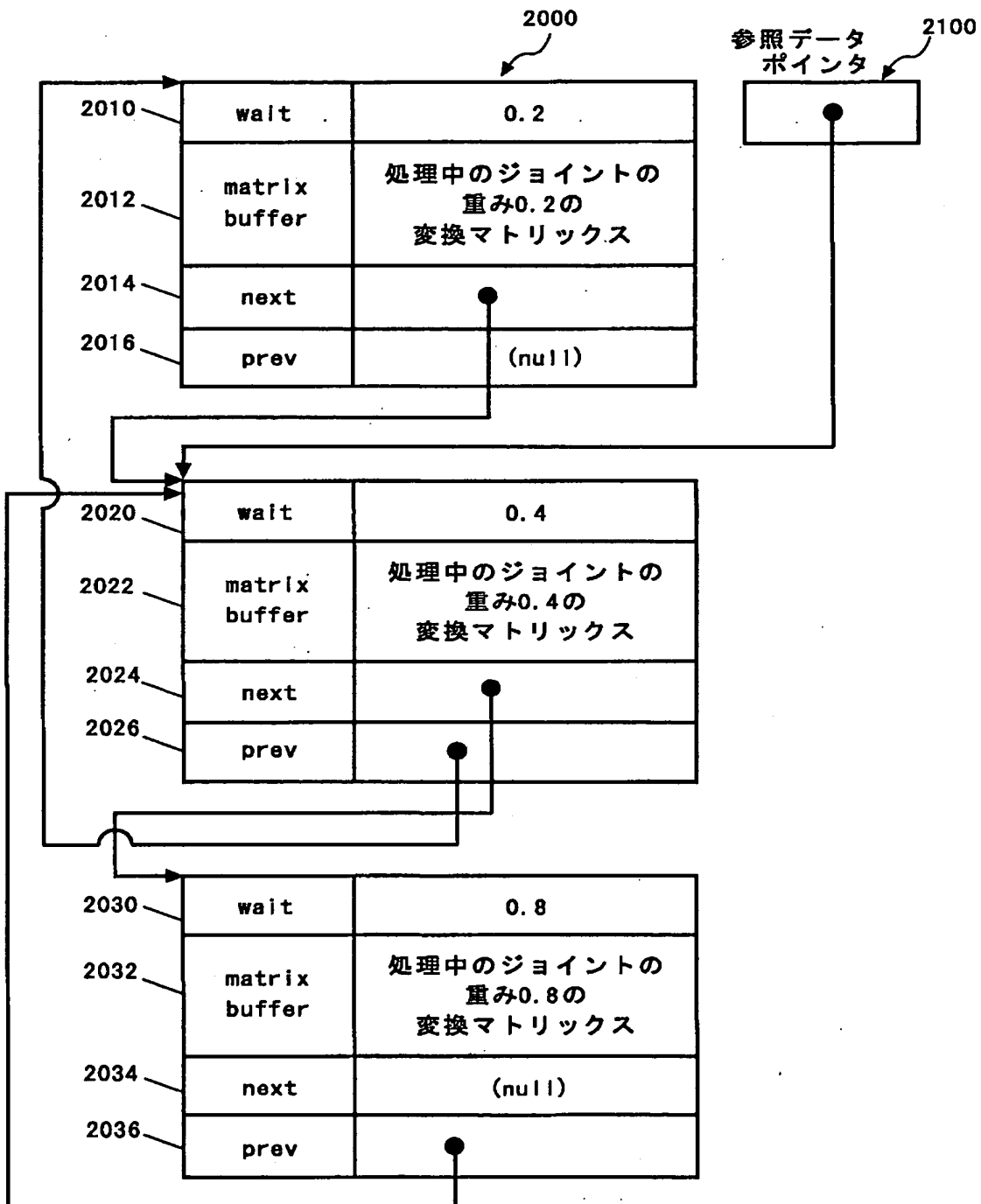
【図 1 5】

910	wait	重み
920	matrix buffer	処理中のジョイントの 重みに応じた 変換マトリックス

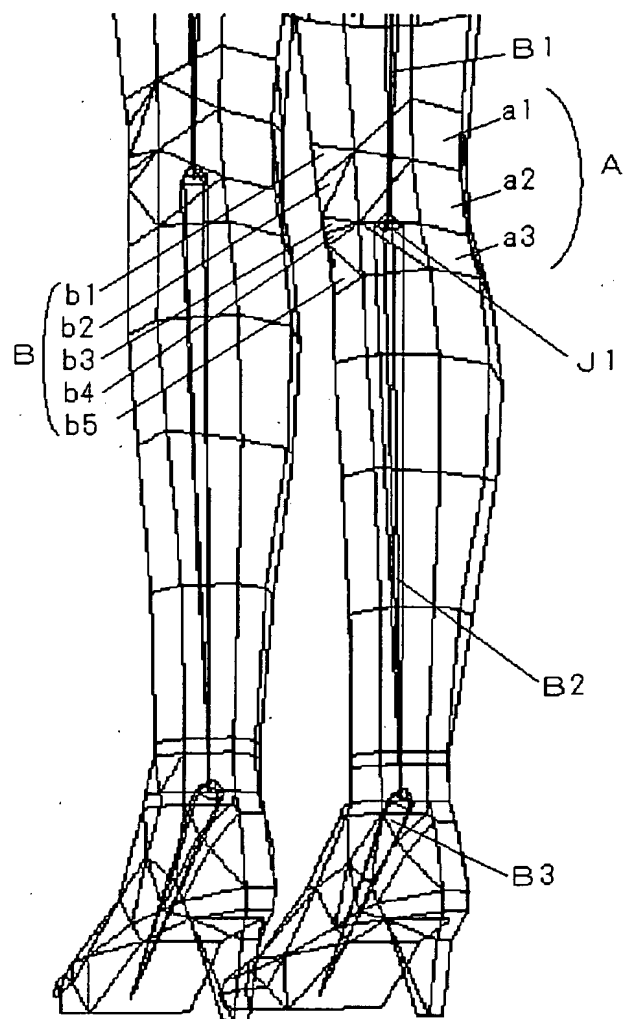
【図 1 6】



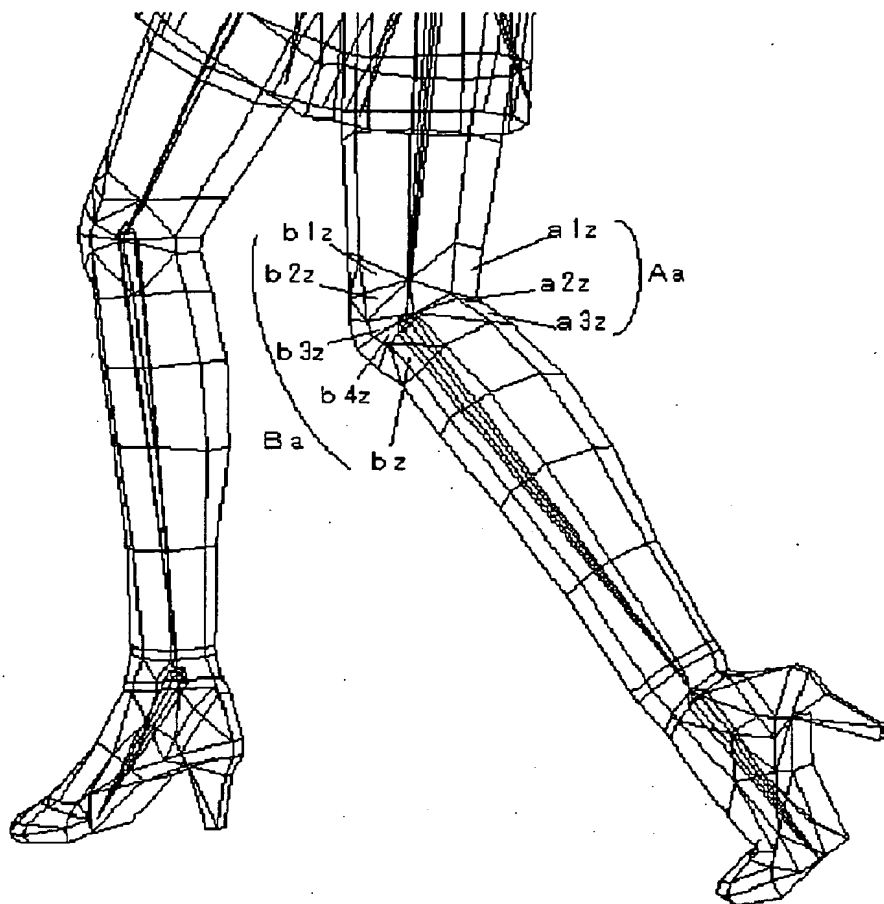
【図 1 7】



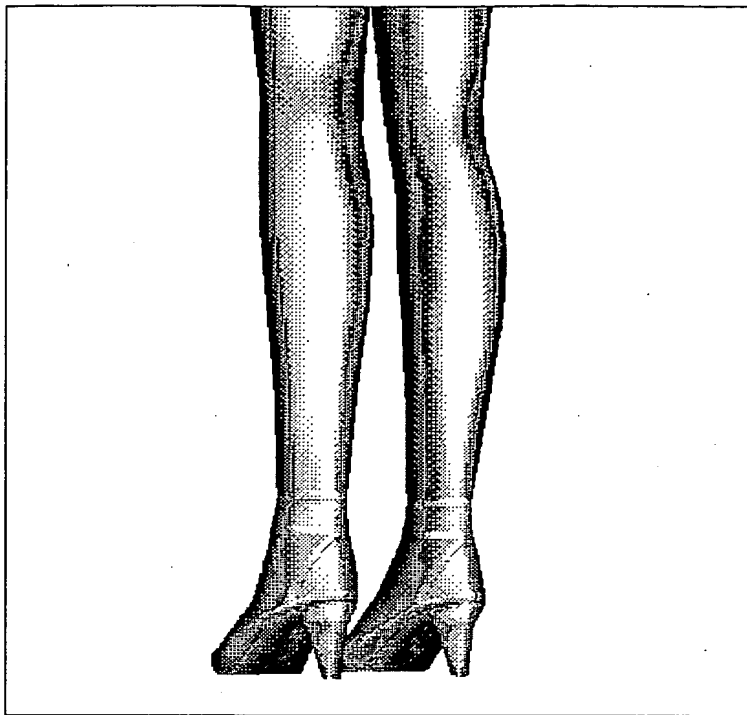
【図18】



【図19】



【図 2 0】



120

【図 2 1】



120

【書類名】 要約書

【要約】

【課題】

頂点数が増加した場合の処理負荷の増加量を少なく抑えた処理で、三次元オブジェクトを滑らかに変形すること。

【解決手段】

未処理ジョイントについてジョイントindex及びジョイントの回転角を取得する(S25,S27)。次に取得されたジョイントindexに対応する未処理の頂点について頂点index及び当該頂点の重み w を取得する(S29-S35)。そして現フレームにおける回転角と重み w に基づいて頂点バッファの座標値を変更する(S37)。ここでは現フレームにおけるジョイントの回転角によるクォータニオン q_1 と単位クォータニオン q_i とを重み w で球状線形補間する。結果のクォータニオン q から本ジョイントにおける変換マトリックス R が求められる。全体の変換マトリックス M は、親ジョイントからの相対座標を表すマトリックス T 、基準回転角を表すマトリックス J 、親ジョイントの変換マトリックス B により、 $M = R J T B$ となる。

【選択図】 図 1 3

認定・付加情報

特許出願の番号	特願2000-019172
受付番号	50000088287
書類名	特許願
担当官	第三担当上席 0092
作成日	平成12年 1月28日

<認定情報・付加情報>

【提出日】 平成12年 1月27日

出 願 人 履 歴 情 報

識別番号

[391049002]

1. 変更年月日 1995年 9月25日

[変更理由] 住所変更

住 所 東京都目黒区下目黒1丁目8番1号

氏 名 株式会社スクウェア